Signal Processing 203 (2023) 108807

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Continuous Encryption Functions for Security Over Networks☆

Yingbo Hua*, Ahmed Maksud

Department of Electrical and Computer Engineering, University of California, Riverside, CA 92521, USA

ARTICLE INFO

Article history:
Received 5 November 2021
Revised 18 July 2022
Accepted 3 October 2022
Available online 5 October 2022

Keywords:
One-way functions
Encryptions
Cancellable passwords
Network security

ABSTRACT

This paper presents a study of continuous encryption functions (CEFs) of secret feature vectors for security over networks, which include physical layer encryption for wireless communications and biometric template security for online Internet applications. While CEFs are defined here to include all prior continuous one-way functions, a good CEF is defined to be a continuous function that turns a random feature vector of limited dimension into a long sequence of numbers in such a way that it is hard to invert and hard to substitute, it has no or little amplification of noise, and its output samples have zero or near-zero correlations and have identical or nearly identical distributions. A number of prior CEFs, such as dynamic random projection, index-of-max hashing and higher-order polynomials, are all shown to fail on these criteria. Based on selected components of singular value decomposition (SVD) of randomly modulated matrices of the feature vector, a family of SVD-CEFs is proposed. Such a SVD-CEF is shown to meet all the criteria for a good CEF and outperform the prior CEFs significantly.

1. Introduction

Encryption is fundamentally important for information security over networks. For a vast range of situations, the amount of user's data far exceeds the amount of secrecy that is available to keep the users' data in complete secrecy. For such a situation, an often called one-way function is required to provide computation based security on top of any given amount of information-theoretic security. The conventional one-way functions are discrete, which in general require a secret key that is 100% reliable.

In this paper, we are interested in applications where a reliable secret key is either not available or insufficient but a limited amount of secrecy is available in some noisy form. One such application is when two separated nodes (Alice and Bob) in a network do not share a secret key but they have their respective estimates of a common physical feature vector (such as reciprocal channel state information). How to use the estimated feature vectors at Alice and Bob to protect a large amount of information transmitted between them is a physical layer encryption problem initially discussed in [1,2] and more recently in [3], which was driven by an interest to protect information transmitted over air against eavesdroppers who may have much stronger channel conditions [5]. Another application is biometric template security for Internet applications [6,7] where network users rely on their own biometric feature vectors for secure online transactions.

The estimated (or measured) feature vectors are always noisy to some degree. To exploit them for encryption, there are two basic approaches. The first is such that Alice and Bob attempt to generate a secret key from their noisy estimates. If successful, this key can be then used to encrypt and decrypt a large amount of information based on a discrete encryption method. But due to noise in the estimated feature vectors, there is no guarantee that the key produced by Alice 100% agrees with the key produced by Bob [14–16]. Any mismatched keys would generally fail a discrete encryption method. Note that an encrypted sequence is typically based on a pseudorandom sequence governed by a seed, i.e., a secret key, and a totally different pseudorandom sequence would be generated with any bit change in the seed. Specifically, if Alice and Bob respectively use their noisy estimates to obtain a pair of limited-length keys $\mathcal{K}_A$ and $\mathcal{K}_B$, then it is likely that $\mathcal{K}_A \neq \mathcal{K}_B$ even though the bit error rate (BER) between $\mathcal{K}_A$ and $\mathcal{K}_B$ can be very small. If Alice and Bob then use $\mathcal{K}_A$ and $\mathcal{K}_B$ respectively via a discrete one-way function to generate a pair of long sequences of pseudorandom sequences $\mathcal{S}_A$ and $\mathcal{S}_B$, then the BER between $\mathcal{S}_A$ and $\mathcal{S}_B$ will be generally very large. Hence $\mathcal{S}_A$ and $\mathcal{S}_B$ cannot be used for encryption and decryption (via modulo addition and subtraction, for example) of a long sequence of information bits.

☆ This work was supported in part by the Army Research Office under Grant Number W911NF-17-1-0581 and the Department of Defense under W911NF-20-2-0267. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation herein.

* Corresponding author.
E-mail addresses: yhua@ece.ucr.edu, yhua@ee.ucr.edu (Y. Hua), ahmed.maksud@email.ucr.edu (A. Maksud).

https://doi.org/10.1016/j.sigpro.2022.108807
0165-1684/

The second approach is what we call here continuous encryption. For physical layer encryption [1,3], for example, a message to be sent by Alice can be encrypted by a continuous encryption function (CEF) based on Alice's estimate of a secret feature vector, and the message can be then recovered by Bob using the same CEF but based on Bob's estimate of the secret feature vector. The noises in the estimated feature vectors in general degrade Bob's recovery of the message but only in a soft or controllable way as long as a signal-to-noise ratio (SNR) of one estimated feature vector relative to the other is high and the CEF has a good enough figure-of-merit (FoM). This second approach is similar in a spirit to many of the methods for biometric template security [6,7].

The contributions of this paper focus on a development of continuous encryption functions (CEFs). We define a CEF as any continuous map of an $N \times 1$ real-valued vector $\mathbf{x}$ onto a (virtually unlimited) long sequence of real-valued numbers: $y_1, y_2, \cdots$. We will denote a CEF by $y_k = f_k(\mathbf{x})$ with $k \geq 1$. If y_k for each k is an arbitrary real-valued number within an interval, we call the CEF type A. If y_k for each k is discrete, we call the CEF type B. A quantization of y_k for all k converts a type-A CEF to a type-B CEF. But not all CEFs have the same quality for applications.

We propose to measure the primary qualities of a CEF $y_k = f_k(\mathbf{x})$ by the following criteria:

1. (Hardness to invert) If $\mathbf{x}$ can be computed (up to a desired precision) from $\{y_k, k \geq 1\}$ with a complexity order that is a polynomial function of N, the CEF is said to be easy (or not hard) to invert. Otherwise, the CEF is said to be hard to invert, which is desired for a good CEF.
2. (Hardness to substitute) If there are such functions g_k that $f_k(\mathbf{x}) = g_k(\mathbf{s}(\mathbf{x}))$ for all $k \geq 1$ where $\mathbf{s}(\mathbf{x})$ is a function of $\mathbf{x}$ and invariant to k, then $\mathbf{s}(\mathbf{x})$ is said to be a substitute input of the CEF. If $\mathbf{s}(\mathbf{x})$ is easy to compute from $\{y_k, k \geq 1\}$, then the CEF is said to be easy to substitute. Otherwise, the CEF is said to be hard to substitute, which is desired for a good CEF.
3. (Sensitivity) A good CEF should be sufficiently responsive to its input but not overly sensitive to small perturbation or noise in its input. The optimal benchmark of the sensitivity to a small perturbation is the sensitivity of a unitary random projection of $\mathbf{x}$. The "noise" referred to in this paper is the difference between two input vectors of interest.
4. (Correlation) Every pair of the output samples of a good CEF should have zero or near-zero correlation if $\mathbf{x}$ has the white Gaussian distribution $\mathcal{N}(0, \sigma_x^2 \mathbf{I}_N)$. If there are strong correlations among the output samples of a CEF, then the CEF is vulnerable to attacks by linear prediction (i.e., y_{k_0} could be estimated by a linear combination of y_k with $k < k_0$).
5. (Invariance) The statistical distribution of y_k for a good CEF should be invariant or nearly invariant to k if $\mathbf{x}$ is of $\mathcal{N}(0, \sigma_x^2 \mathbf{I}_N)$. One benefit from the invariance is that it makes quantization of y_k for all k easier (i.e., a good quantizer for y_{k_0} would be equally good for y_k for all $k \neq k_0$).

If a CEF meets all of the above criteria, the CEF is said to be a good CEF. A good type-A CEF can be viewed as a generator of quasi-continuous pseudorandom numbers (QPRNs). These QCPRNs are based on a continuous feature vector $\mathbf{x}$ as its "seed", which is different from the traditional PRN generators that rely on discrete seed.

It seems not possible to prove whether a CEF is hard to invert or hard to substitute although one can try to prove that a CEF is not hard to invert or not hard to substitute. This is an open problem similar to that of discrete one-way functions [17,18,23] even though the use of discrete one-way functions in practice is indispensable. We will say that a CEF is empirically hard to attack if there is a strong empirical evidence suggesting that the CEF is hard to invert and hard to substitute. As for sensitivity, correlation and invariance of a CEF, one can apply statistical analysis and/or computer simulation to quantify the degree to which these criteria are satisfied by the CEF.

The family of CEFs includes all prior hard-to-invert (i.e., one-way) continuous functions proposed in the literature. The hard-to-invert property is widely desired in applications. The hard-to-substitute property is also important for a similar reason. If an attacker is able to determine a substitute input from a prior exposure of y_k for $1 \leq k \leq K_0$, then all future output samples y_k for $k > K_0$ can be predicted by the attacker. It is clear that "easy to invert" implies "easy to substitute", but the reverse is not true in general. We say that a CEF is easy to attack if it is easy to invert or easy to substitute. Equivalently, a CEF is said to be hard to attack if it is hard to invert and hard to substitute.

The sensitivity of a CEF to noise is clearly important in applications. The optimal sensitivity is that of a unitary random projection as discussed later in this paper. To have a small noise sensitivity (relative to the optimal), a CEF must be locally continuous with probability one subject to a continuous randomness of $\mathbf{x}$. For a type-A CEF, we can measure its sensitivity by a FoM such as the square-rooted ratio of $\mathtt{SNR}_{\mathbf{x}}$ over $\mathtt{SNR}_{\mathbf{y}}$ where $\mathtt{SNR}_{\mathbf{x}}$ and $\mathtt{SNR}_{\mathbf{y}}$ are some signal-to-noise ratios (SNRs) of $\mathbf{x}$ and $\mathbf{y} = [y_1, \cdots, y_K]^T$ respectively, e.g., see (62) later. The optimal desired value of such a FoM is one. For a type-B CEF, the sensitivity can be measured by BER in y_k for $k \geq 1$ caused by random perturbations in $\mathbf{x}$, which will be discussed in detail in Section 7.

The output correlation of a CEF is also important. For example, if a CEF has a "zero sensitivity to noise", then its output would be a constant with perfect correlations, which is obviously a useless CEF. In general, nonzero correlations among the output samples of a CEF would allow attacks by linear prediction, which is not desirable. So, a good CEF should have zero or near zero output correlations. The invariance of the output distribution of a CEF is also desirable especially for the purpose of quantization. The correlation and invariance properties of a proposed CEF will be discussed in detail in this paper.

1.1. Prior works and current contributions

It appears that the prior CEFs all exploit (or can all exploit) any available secret key S (as the seed) to produce pseudorandom numbers or operations needed in the functions. A method to invert such a CEF in general has a complexity order equal to $C_{N,M} 2^{N_S}$, where N_S is the number of binary bits in the secret key, and $C_{N,M}$ is the complexity to invert the CEF if the secret key is given. Unless mentioned otherwise, we will refer to $C_{N,M}$ as the complexity of attack. A good understanding of $C_{N,M}$ is important for situations where N_S is not sufficiently large or simply zero.

The random projection (RP) method in [8] and the dynamic random projection (DRP) method in [9] are type-A CEFs before a quantization is applied at the last step of the functions. The Index-of-Maximum (IoM) hashing in [11] is inherently a type-B CEF. The higher-order polynomials (HOP) in [10] are a type-A CEF.

We will show that for the RP method, the DRP method and the IoM algorithm 1, $C_{N,M} = P_{N,M}$ with $P_{N,M}$ denoting a polynomial function of both N and M; and for the IoM algorithm 2, $C_{N,M} = L_{N,M} 2^N$ with $L_{N,M}$ being a linear function of N and M respectively. The HOP method is shown to be easy to substitute. There are two versions of DRP based on function-I and function-II in [9]. Unless mentioned otherwise, we will refer to the function-II version by DRP. We will also show that HOP is highly sensitive to noise, and the output samples of RP, DRP and IoM all have a high peak correlation.

Another major contribution of this paper is a new family of nonlinear CEFs called SVD-CEF. This family of CEFs is of type A and based on selected components of singular value decomposition

Table 1
Comparison of CEFs in the absence of secret key.

	Ref	Type	Comp.	H.I.	H.S.	Attack C.	Cor.	Sen.	Inv.
RP	[8]	A	$\mathcal{O}(N)$	No	-	$P_{N,M}$	Bad	-	-
DRP	[9]	A	$\mathcal{O}(N)$	No	-	$P_{N,M}$	Bad	-	-
URP	Here	A	$\mathcal{O}(N)$	No	-	$P_{N,M}$	Bad	Best	Best
HOP	[10]	A	$\mathcal{O}(N)$	-	No	$P_{N,M}$	-	Bad	-
IoM-1	[11]	B	$\mathcal{O}(N^2)$	No	-	$L_{N,M}$	Bad	-	-
IoM-2	[11]	B	$\mathcal{O}(N^2)$	Yes	Yes	$L_{N,M}2^N$	Bad	Not as good	-
SVD-CEF	Here	A	$\mathcal{O}(N^3)$	Yes	Yes	$P_{N,M}2^{\zeta N}$	Good	Good	Good

(SVD) of randomly modulated matrices of $\mathbf{x}$. Based on the empirical evidences shown in this paper, the complexity order to attack a SVD-CEF is $C_{N,M} = P_{N,M}2^{\zeta N}$ where $\zeta > 1$ is typically substantially larger than one and increases as N increases. We will show that the output of SVD-CEF also has good properties in terms of noise sensitivity, output correlation and distribution invariance. Furthermore, we will show that a quantized SVD-CEF outperforms the IoM algorithm 2 dramatically in terms of BER. Additional comparison of SVD-CEF with other methods is available in [4] where CEF is applied for secret key generation.

Table 1 provides a summary comparison of CEFs discussed in this paper, where each entry in the "Comp." column is the order of the forward computational complexity per output sample of the CEF in terms of N, "Yes" in the H.I. column means "empirically hard to invert", "No" in the H.I. column "not hard to invert", "Yes" in the H.S. column "empirically hard to substitute", "No" in the H.S. column "not hard to substitute", and the column of "Attack C." shows the attack complexity $C_{N,M}$. The columns of "Cor., Sen. and Inv." correspond to "correlation, sensitivity and invariance" respectively. An entry marked as "-" is an entry that is not very important due to "No" or "Bad" in another column. But an entry that has the optimal performance is marked as "Best". The two entries of "Best" in the table are easy to prove. The entries of "Good", "Not as good", "Bad", "Yes" and "No" are established via analysis and simulation shown in this paper. The complexity orders shown in the table are also detailed in this paper.

As shown in this paper, SVD-CEF stands out as a good CEF as measured by the five criteria shown earlier. A main reason why SVD-CEF is hard to invert and hard to substitute is that the components of SVD of a randomly modulated matrix of the secret vector $\mathbf{x}$ are nonlinearly related to $\mathbf{x}$. More specifically, given the output samples of SVD-CEF, finding $\mathbf{x}$ or its substitute amounts to finding the solution of a set of multivariate second-order polynomials. A main reason why SVD-CEF yields uncorrelated output samples is also because of a highly nonlinear relationship between the output samples of SVD-CEF and $\mathbf{x}$. See discussion of equation (3) in [4].

1.2. The rest of the paper

In Section 2, we review a linear family of CEFs, including RP and DRP. We will also discuss a unitary random projection (URP) and a transformation from the N-dimensional real space $\mathcal{R}^N$ to the N-dimensional sphere of unit radius $\mathcal{S}^N(1)$. The URP would be an ideal CEF if there is a (strong) secret key shared by Alice and Bob. But if there is no (strong) secret key, URP has the weakness of being easy to invert and having high output correlations as highlighted later in this paper. In Section 3, we review a family of nonlinear CEFs, including HOP and IoM. In Section 4, we present a new family of nonlinear CEFs called SVD-CEF, which is a new development from our prior works in [1,2]. In Section 5, we provide empirical details to explain why SVD-CEF is hard to attack. In Section 6, we provide statistical analyses of SVD-CEF as well as simulation results to show why SVD-CEF has good properties in terms of sensitivity, correlation and invariance. In Section 7, we show a detailed comparison of the noise sensitivities of a quantized SVD-CEF and the IoM algorithm 2, which shows a significant advantage of SVD-CEF. The conclusion is given in Section 8. A previous version of this paper is posted at [24].

2. Linear family of CEFs

A family of linear CEFs can be expressed as follows:

$$\mathbf{y} = \mathbf{R}_S\mathbf{x} \tag{1}$$

where $\mathbf{y} = [y_1, y_2, \cdots, y_M]^T$, M is a large integer, $\mathbf{R}_S$ is a $M \times N$ pseudorandom matrix dependent on a secret key S. Let the ith $M_i \times 1$ subvector of $\mathbf{y}$ be $\mathbf{y}_i$, and the ith $M_i \times N$ block matrix of $\mathbf{R}_S$ be $\mathbf{R}_{S,i}$. Then it follows that

$$\mathbf{y}_i = \mathbf{R}_{S,i}\mathbf{x} \tag{2}$$

where $i = 1, \cdots, I$ and $\sum_{i=1}^{I} M_i = M$.

2.1. Random projection

The linear family of CEFs includes the random projection (RP) method shown in [8] and applied in [12]. If S is known, so is $\mathbf{R}_{S,i}$ for all i. If $\mathbf{y}_i$ for some i is known/exposed and $\mathbf{R}_{S,i}$ is of the full column rank N, then $\mathbf{x}$ is given by $\mathbf{R}_{S,i}^+\mathbf{y}_i = (\mathbf{R}_{S,i}^T\mathbf{R}_{S,i})^{-1}\mathbf{R}_{S,i}^T\mathbf{y}_i$ where $^+$ denotes pseudo-inverse. If $\mathbf{R}_{S,i}$ is not of full column rank, then $\mathbf{x}$ can be computed from a set of outputs like (for example) $\mathbf{y}_1, \cdots, \mathbf{y}_L$ where L is such that the vertical stack of $\mathbf{R}_{S,1}, \cdots, \mathbf{R}_{S,L}$, denoted by $\mathbf{R}_{S,1:L}$, is of the full column rank N.

If S is unknown, then a method to compute $\mathbf{x}$ includes a discrete search for the N_S bits of S as follows

$$\min_S \min_{\mathbf{x}} \|\mathbf{y}_{1:L} - \mathbf{R}_{S,1:L}\mathbf{x}\| = \min_S \|\mathbf{y}_{1:L} - \mathbf{R}_{S,1:L}\mathbf{R}_{S,1:L}^+\mathbf{y}_{1:L}\| \tag{3}$$

where $\mathbf{y}_{1:L}$ is the vertical stack of $\mathbf{y}_1, \cdots, \mathbf{y}_L$. The total complexity of the above attack algorithm with unknown key S is $P_{N,M}2^{N_S}$ with $P_{N,M}$ being a linear function of $\sum_{i=1}^{L} M_i$ and a cubic function of N.

So, RP is not hard to attack (subject to a small N_S).

2.2. Dynamic random projection

The dynamic random projection (DRP) method proposed in [9] and also discussed in [7] can be described by

$$\mathbf{y}_i = \mathbf{R}_{S,i,\mathbf{x}}\mathbf{x} \tag{4}$$

where $\mathbf{R}_{S,i,\mathbf{x}}$ is the ith realization of a random matrix that depends on both S and $\mathbf{x}$. Since $\mathbf{R}_{S,i,\mathbf{x}}$ is discrete, $\mathbf{y}_i$ in (4) is a *locally* linear function of $\mathbf{x}$. (There is a nonzero probability that a small perturbation $\mathbf{w}$ in $\mathbf{x}' = \mathbf{x} + \mathbf{w}$ leads to $\mathbf{R}_{S,i,\mathbf{x}'}$ being substantially different from $\mathbf{R}_{S,i,\mathbf{x}}$. This is not a desirable outcome for biometric templates although the probability may be small.) Two methods were proposed in [9] to construct $\mathbf{R}_{S,i,\mathbf{x}}$, which were called "Functions I and II" respectively. For simplicity of notation, we will now suppress i and S in (4) and write it as

$$\mathbf{y} = \mathbf{R}_{\mathbf{x}}\mathbf{x} \tag{5}$$

2.2.1. Assuming "Function I" in [9]

In this case, the ith element of $\mathbf{y}$, denoted by v_i, corresponds to the ith slot shown in [9] and can be written as

$$v_i = \mathbf{r}_{x,i}^T \mathbf{x} \tag{6}$$

where $\mathbf{r}_{x,i}^T$ is the ith row of $\mathbf{R_x}$. But $\mathbf{r}_{x,i}^T$ is one of L key-dependent pseudorandom vectors $\mathbf{r}_{i,1}^T, \cdots, \mathbf{r}_{i,L}^T$ that are independent of $\mathbf{x}$ and known if S is known. So we can also write

$$v_i = \mathbf{r}_i^T \bar{\mathbf{x}} \tag{7}$$

where $\mathbf{r}_i^T = [\mathbf{r}_{i,1}^T, \cdots, \mathbf{r}_{i,L}^T]^T$, and $\bar{\mathbf{x}} \in \mathcal{R}^{LN}$ is a sparse vector consisting of zeros and $\mathbf{x}$. Before $\mathbf{x}$ is known, the position of $\mathbf{x}$ in $\bar{\mathbf{x}}$ is initially unknown.

If an attacker has stolen K realizations of v_i (denoted by $v_{i,1}, \cdots, v_{i,K}$), then it follows that

$$\mathbf{v}_i = \mathbf{R}_i \bar{\mathbf{x}} \tag{8}$$

where $\mathbf{v}_i = [v_{i,1}, \cdots, v_{i,K}]^T$, and $\mathbf{R}_i$ is the vertical stack of K key-dependent random realizations of $\mathbf{r}_i^T$. With $K \geq LN$, $\mathbf{R}_i$ is of the full column rank LN with probability one, and in this case the above equation (when given the key S) is linearly invertible with a complexity order equal to $\mathcal{O}((LN)^3)$.

An even simpler method of attack is as follows. Since $v_{i,k} = \mathbf{r}_{i,k,l}^T \mathbf{x}$ where $l \in \{1, \cdots, L\}$ and $\mathbf{r}_{i,k,l}$ for all i, k and l are known, then we can compute

$$\begin{aligned} l^* &= arg \min_{l \in \{1,\cdots,L\}} \min_{\mathbf{x}} \|\mathbf{v}_i - \mathbf{R}_{i,l}\mathbf{x}\|^2 \\ &= arg \min_{l \in \{1,\cdots,L\}} \|\mathbf{v}_i - \mathbf{R}_{i,l}\mathbf{R}_{i,l}^+ \mathbf{v}_i\|^2 \end{aligned} \tag{9}$$

where $\mathbf{R}_{i,l}$ is the vertical stack of $\mathbf{r}_{i,k,l}^T$ for $k = 1, \cdots, K$. Provided $K \geq N$, $\mathbf{R}_{i,l}$ has the full column rank with probability one. In this case, the correct solution of $\mathbf{x}$ is given by $\mathbf{R}_{i,l^*}^+ \mathbf{v}_i$. This method has a complexity order equal to $\mathcal{O}(LN^3)$.

2.2.2. Assuming "Function II" in [9]

To attack "Function II" with known S, it is equivalent to consider the following signal model:

$$v_k = \sum_{n=1}^{N} r_{k,l_k,n} x_n \tag{10}$$

where v_k is available for $k = 1, \cdots, K$, $r_{k,l,n}$ for $1 \leq k \leq K$, $1 \leq l \leq L$ and $1 \leq n \leq N$ are random but known[1] numbers (when given S), x_n for all n are unknown, and l_k is a k-dependent random/unknown choice from $[1, \cdots, L]$.

We can write

$$\mathbf{v} = \mathbf{R}\mathbf{x} \tag{11}$$

where $\mathbf{v}$ is a stack of all v_k, $\mathbf{x}$ is a stack of all x_n, and $\mathbf{R}$ is a stack of all $r_{k,l_k,n}$ (i.e., $(\mathbf{R})_{k,n} = r_{k,l_k,n}$). In this case, $\mathbf{R}$ is a random and unknown choice from L^K possible known matrices. An exhaustive search would require the $\mathcal{O}(L^K)$ complexity with $K \geq N+1$.

Now we consider a different approach of attack. Since $r_{k,l,n}$ for all k, l, n are known, we can compute

$$c_{n,n'} = \frac{1}{KL} \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{l'=1}^{L} r_{k,l,n} r_{k,l',n'} \tag{12}$$

If $r_{k,l,n}$ are pseudo i.i.d. random (but known) numbers of zero mean and variance one, then for large K (e.g., $K \gg L^2$) we have $c_{n,n'} \approx \delta_{n,n'}$.

Also define

$$y_n = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} v_k r_{k,l,n} = \sum_{n'=1}^{N} \hat{c}_{n,n'} x_{n'} \tag{13}$$

where $n = 1, \cdots, N$ and

$$\hat{c}_{n,n'} = \frac{1}{K} \sum_{k=1}^{K} \sum_{l=1}^{L} r_{k,l,n} r_{k,l_k,n'}. \tag{14}$$

If $r_{k,l,n}$ are i.i.d. of zero mean and unit variance, then for large K we have $\hat{c}_{n,n'} \approx c_{n,n'} \approx \delta_{n,n'}$ and hence

$$y_n \approx x_n. \tag{15}$$

More generally, if we have $\hat{c}_{n,n'} \approx c_{n,n'}$ with a large K, then

$$\mathbf{y} \approx \mathbf{C}\mathbf{x} \tag{16}$$

where $(\mathbf{y})_n = y_n$, and $(\mathbf{C})_{n,n'} = c_{n,n'}$. Hence,

$$\mathbf{x} \approx \mathbf{C}^{-1}\mathbf{y}. \tag{17}$$

With an initial estimate $\hat{\mathbf{x}}$ of $\mathbf{x}$, we can then do the following to refine the estimate:

1. For each of $k = 1, \cdots, K$, compute $l_k^* = arg\min_{l \in [1,\cdots,L]} |v_k - \sum_{n=1}^{N} r_{k,l,n}\hat{x}_n|$.
2. Recall $\mathbf{v} = \mathbf{R}\mathbf{x}$. But now use $(\mathbf{R})_{k,n} = r_{k,l_k^*,n}$ for all k and n, and replace $\hat{\mathbf{x}}$ by
$$\hat{\mathbf{x}} = (\mathbf{R}^T\mathbf{R})^{-1}\mathbf{R}^T\mathbf{v} \tag{18}$$
3. Go to step 1 until convergence.

Note that all entries in $\mathbf{R}$ are discrete. Once the correct $\mathbf{R}$ is found, the exact $\mathbf{x}$ is obtained. The above algorithm converges to either the exact $\mathbf{x}$ or a wrong $\mathbf{x}$. But with a sufficiently large K with respect to a given pair of N and L, our simulation shows that above attack algorithm yields the exact $\mathbf{x}$ with high probabilities. For example, for $N = 8$, $L = 8$ and $K = 23L$, the successful rate is 99%. And for $N = 16$, $L = 48$ and $K = 70L$, the successful rate is 98%. In the experiment, for each set of N, L and K, 100 independent realizations of all elements in $\mathbf{x}$ and $\mathbf{R}$ were chosen from i.i.d. Gaussian distribution with zero mean and unit variance, i.e., $\mathcal{N}(0,1)$. The successful rate was based on the 100 realizations.

In [9], an element-wise quantized version of $\mathbf{v}$ was further suggested to improve the hardness to invert. In this case, the vector potentially exposable to an attacker can be written as

$$\hat{\mathbf{v}} = \mathbf{R}\mathbf{x} + \mathbf{w} \tag{19}$$

where $\mathbf{w}$ can be modelled as a white noise vector uncorrelated with $\mathbf{R}\mathbf{x}$. The above attack algorithm with $\mathbf{v}$ replaced by $\hat{\mathbf{v}}$ also applies although a larger K is needed to achieve the same rate of successful attack.

In all of the above cases, the computational complexity for a successful attack is a polynomial function of N, L and/or K when the secret key S is given.

2.3. Unitary Random Projection (URP)

None of the RP and DRP methods is homomorphic. To have a homomorphic CEF whose input and output have the same distance measure, we can use

$$\mathbf{y}_k = \mathbf{Q}_k \mathbf{x} \tag{20}$$

where $\mathbf{Q}_k \in \mathcal{R}^{N \times N}$ for each realization index k is a pseudorandom unitary matrix governed by a secret key S. One way to generate $\mathbf{Q}_k$ is to compute the QR decomposition [19] of a random matrix $\mathbf{X}_k$ whose entries are pseudorandom numbers (including Gaussian

[1] "random but known" means "known" strictly speaking despite a pseudorandomness.

random numbers) from a standard cryptographically secure pseudorandom number generator. It is important to note that if there is a secret key with its length $N_S \geq N$, then URP is also hard to invert strictly speaking. But as stressed earlier, this paper focuses on the case where $N_S \ll N$ or simply $N_S = 0$.

Let $\mathbf{x}' = \mathbf{x} + \mathbf{w}$ with $\mathbf{w}$ being a noise. Then $\mathbf{y}'_k \doteq \mathbf{Q}_k\mathbf{x}' = \mathbf{Q}_k\mathbf{x} + \mathbf{Q}_k\mathbf{w}$. It follows that the SNR of $\mathbf{y}'_k$ equals the SNR of $\mathbf{x}'$, and hence the FoM of URP equals one. We can view the noise sensitivity of URP as optimal. In fact, if Alice and Bob do share a strong secret key, then the URP would be an ideal CEF as it would meet perfectly all the five criteria. However, like RP and DRP, URP is easy to attack if the secret key is weak or does not exist. Furthermore, as shown later, without a secret key or equivalently with a known set of $\mathbf{Q}_k$ for all k, the output samples of URP are highly correlated with each other.

Note that each of the linear CEFs requires a forward per-sample computation complexity equal to $\mathcal{O}(N)$. For example, to produce N output samples of URP, we need to generate the $N \times N$ unitary matrix $\mathbf{Q}_k$, which requires a computational complexity equal to $\mathcal{O}(N^2)$. We also need to compute the product $\mathbf{Q}_k\mathbf{x}$ which costs another $\mathcal{O}(N^2)$. So, the per-sample complexity is $\mathcal{O}(N)$.

If $\mathbf{x}$ consists of i.i.d. $\mathcal{N}(0, \sigma_x^2)$, all entries of $\mathbf{y}_i$ for all i are also $\mathcal{N}(0, \sigma_x^2)$, which is a desired invariance of statistical distribution. But the entries of $\mathbf{y}_i$ in general have significant correlations with entries of $\mathbf{y}_j$ for $j \neq i$ (even though the N entries of $\mathbf{y}_i$ for each i have zero correlations among themselves). Simulation results on the correlations of RP, DRP and URP will be shown later.

2.3.1. Transformation from $\mathcal{R}^N$ to $\mathcal{S}^N(1)$

For URP, $\|\mathbf{y}_k\| = \|\mathbf{x}\|$, which means that $\|\mathbf{x}\|$ is readily available from $\mathbf{y}_i$. If $\|\mathbf{x}\|$ needs some protection from an exposed $\mathbf{y}_i$, we can apply the transformation shown next.

We now introduce a transformation from the N-dimensional vector space $\mathcal{R}^N$ to the N-dimensional sphere of unit radius $\mathcal{S}^N(1)$. Let $\mathbf{x} \in \mathcal{R}^N$. Define

$$\mathbf{v} = \begin{bmatrix} \frac{1}{\|\mathbf{x}\|\sqrt{1+\|\mathbf{x}\|^2}}\mathbf{x} \\ \frac{\|\mathbf{x}\|}{\sqrt{1+\|\mathbf{x}\|^2}} \end{bmatrix} \tag{21}$$

which clearly satisfies $\mathbf{v} \in \mathcal{S}^N(1)$. Then, we let

$$\mathbf{y}_k = \mathbf{Q}_k\mathbf{v} \tag{22}$$

where $\mathbf{Q}_k$ is now a $(n+1) \times (n+1)$ unitary random matrix governed by a secret key S.

Let $\mathbf{y}'_k = \mathbf{R}_k\mathbf{v}'$. It follows that $\|\mathbf{y}'_k - \mathbf{y}_k\| = \|\mathbf{v}' - \mathbf{v}\|$. But since $\mathbf{v}$ is now a nonlinear function of $\mathbf{x}$, the relationship between $\|\mathbf{v}' - \mathbf{v}\|$ and $\|\mathbf{x}' - \mathbf{x}\|$ is more complicated, which we discuss below.

Let us consider $\mathbf{x}' = \mathbf{x} + \mathbf{w}$. One can verify that

$$\begin{aligned}\|\mathbf{v}' - \mathbf{v}\| &= \left\| \begin{bmatrix} \frac{\mathbf{x}+\mathbf{w}}{\|\mathbf{x}+\mathbf{w}\|\sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2}} \\ \frac{\|\mathbf{x}+\mathbf{w}\|}{\sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2}} \end{bmatrix} - \begin{bmatrix} \frac{\mathbf{x}}{\|\mathbf{x}\|\sqrt{1+\|\mathbf{x}\|^2}} \\ \frac{\|\mathbf{x}\|}{\sqrt{1+\|\mathbf{x}\|^2}} \end{bmatrix} \right\| \\ &= \left\| \begin{bmatrix} \frac{\mathbf{a}}{b} \\ \frac{c}{d} \end{bmatrix} \right\| \end{aligned} \tag{23}$$

where

$$\begin{aligned}\mathbf{a} &= (\mathbf{x}+\mathbf{w}) \cdot \|\mathbf{x}\| \cdot \sqrt{1+\|\mathbf{x}\|^2} \\ &\quad - \mathbf{x} \cdot \|\mathbf{x}+\mathbf{w}\| \cdot \sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2}\end{aligned} \tag{24}$$

$$b = \|\mathbf{x}\| \cdot \sqrt{1+\|\mathbf{x}\|^2} \cdot \|\mathbf{x}+\mathbf{w}\| \cdot \sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2} \tag{25}$$

$$c = \|\mathbf{x}+\mathbf{w}\| \cdot \sqrt{1+\|\mathbf{x}\|^2} - \|\mathbf{x}\| \cdot \sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2} \tag{26}$$

$$d = \sqrt{1+\|\mathbf{x}\|^2\|} \cdot \sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2}. \tag{27}$$

To derive a simpler relationship between $\|\mathbf{v}' - \mathbf{v}\|$ and $\|\mathbf{x}' - \mathbf{x}\| = \|\mathbf{w}\|$, we will assume $\|\mathbf{w}\| \ll r = \|\mathbf{x}\|$ and apply the first order approximations. Also we can write

$$\mathbf{w} = \eta_x\mathbf{w}_x + \eta_\perp\mathbf{w}_\perp \tag{28}$$

where $\mathbf{w}_x$ is a unit-norm vector in the direction of $\mathbf{x}$, and $\mathbf{w}_\perp$ is a unit-norm vector orthogonal to $\mathbf{x}$. Then,

$$\|\mathbf{w}\|^2 = \eta_x^2 + \eta_\perp^2 \tag{29}$$

$$\mathbf{x}^T\mathbf{w} = \eta_x\|\mathbf{x}\| = \eta_x r. \tag{30}$$

It follows that

$$\begin{aligned}\|\mathbf{x}+\mathbf{w}\| &\approx \|\mathbf{x}\| \\ &\quad + \frac{1}{2\|\mathbf{x}\|}(\|\mathbf{w}\|^2 + 2\mathbf{x}^T\mathbf{w}) \\ &= r + \frac{1}{2r}(\eta_x^2 + \eta_\perp^2 + 2r\eta_x) \\ &\approx r + \frac{1}{2r}(\eta_\perp^2 + 2r\eta_x)\end{aligned} \tag{31}$$

$$\begin{aligned}\sqrt{1+\|\mathbf{x}+\mathbf{w}\|^2} &\approx \sqrt{1+\|\mathbf{x}\|^2} \\ &\quad + \frac{1}{2\sqrt{1+\|\mathbf{x}\|^2}}(\|\mathbf{w}\|^2 + 2\mathbf{x}^T\mathbf{w}) \\ &\approx \sqrt{1+r^2} + \frac{1}{2\sqrt{1+r^2}}(\eta_\perp^2 + 2r\eta_x).\end{aligned} \tag{32}$$

Then, one can verify that

$$\mathbf{a} \approx \mathbf{w}r\sqrt{1+r^2} - \mathbf{x}\frac{1}{2}\left(\frac{r}{\sqrt{1+r^2}} + \frac{\sqrt{1+r^2}}{r}\right)(\eta_\perp^2 + 2r\eta_x) \tag{33}$$

and

$$\begin{aligned}\|\mathbf{a}\|^2 &= r^2(1+r^2)(\eta_x^2 + \eta_\perp^2) \\ &\quad + \frac{1}{4}r^2\left(\frac{r}{\sqrt{1+r^2}} + \frac{\sqrt{1+r^2}}{r}\right)^2(\eta_\perp^2 + 2r\eta_x)^2 \\ &\quad - \eta_x r^2\sqrt{1+r^2}\left(\frac{r}{\sqrt{1+r^2}} + \frac{\sqrt{1+r^2}}{r}\right)(\eta_\perp^2 + 2r\eta_x) \\ &\approx r^2(1+r^2)(\eta_x^2 + \eta_\perp^2) \\ &\quad + r^4\left(\frac{r}{\sqrt{1+r^2}} + \frac{\sqrt{1+r^2}}{r}\right)^2\eta_x^2 \\ &\quad - 2r^3\sqrt{1+r^2}\left(\frac{r}{\sqrt{1+r^2}} + \frac{\sqrt{1+r^2}}{r}\right)\eta_x^2 \\ &= r^2(1+r^2)\eta_\perp^2 + \frac{r^6}{1+r^2}\eta_x^2\end{aligned} \tag{34}$$

where the approximations hold because of $\eta_x \ll r$ and $\eta_\perp \ll r$. Similarly, we have

$$b^2 \approx r^4(1+r^2)^2 \tag{35}$$

$$c^2 \approx \left(\frac{1}{2r\sqrt{1+r^2}}(\eta_\perp^2 + 2r\eta_x)\right)^2 \approx \frac{1}{(1+r^2)}\eta_x^2 \tag{36}$$

$$d^2 \approx (1+r^2)^2. \tag{37}$$

Hence

$$\|\mathbf{v}' - \mathbf{v}\|^2 = \frac{\|\mathbf{a}\|^2}{b^2} + \frac{c^2}{d^2} \approx \frac{1}{r^2(1+r^2)}\eta_\perp^2 + \frac{r^2+1}{(1+r^2)^3}\eta_x^2. \tag{38}$$

It is somewhat expected that the larger is r, the less are the sensitivities of $\|\mathbf{v}' - \mathbf{v}\|^2$ to $\eta_\perp$ and η_x. But the sensitivities of

$\|\mathbf{v}'-\mathbf{v}\|^2$ to $\eta_\perp$ and η_x are different in general, which also vary differently as r varies. If $r \ll 1$, then

$$\|\mathbf{v}'-\mathbf{v}\|^2 \approx \frac{1}{r^2}\eta_\perp^2 + \eta_x^2 \tag{39}$$

which shows a higher sensitivity of $\|\mathbf{v}'-\mathbf{v}\|^2$ to $\eta_\perp$ than to η_x. If $r \gg 1$, then

$$\|\mathbf{v}'-\mathbf{v}\|^2 \approx \frac{1}{r^4}\eta_\perp^2 + \frac{1}{r^4}\eta_x^2 = \frac{1}{r^4}\|\mathbf{w}\|^2 \tag{40}$$

which shows equal sensitivities of $\|\mathbf{v}'-\mathbf{v}\|^2$ to $\eta_\perp$ and η_x respectively.

The above results show how $\|\mathbf{v}'-\mathbf{v}\|^2$ changes with $\mathbf{w} = \eta_\perp \mathbf{w}_\perp + \eta_x \mathbf{w}_x$ subject to $\|\mathbf{w}\| \ll \|\mathbf{x}\| = r$ or equivalently $\sqrt{\eta_\perp^2 + \eta_x^2} \ll r$.

For larger $\|\mathbf{w}\|$, the relationship between $\|\mathbf{v}'-\mathbf{v}\|^2$ and $\|\mathbf{w}\|$ is not as simple. But one can verify that if $\|\mathbf{w}\| \gg r \gg 1$, then $\|\mathbf{v}'-\mathbf{v}\| \approx 1/r$.

3. Nonlinear family of CEFs

If the secret key S available is not large enough, then we will need a CEF that is hard to attack even if S is known. Such a CEF has to be nonlinear.

3.1. Higher-order polynomials

A family of higher-order polynomials (HOP) was suggested in [10] as a hard-to-invert continuous function. But we show here that HOP is not hard to substitute. Let $\mathbf{y} = [y_1, \cdots, y_M]^T$ and $\mathbf{x} = [x_1, \cdots, x_N]^T$ where y_m is a HOP of $x_1, \cdots, x_N$ with pseudorandom coefficients. Namely, $y_m = f_m(x_1, \cdots, x_N) = \sum_{j=0}^{J} c_{m,j} \prod_{i=1}^{N} x_i^{p_{i,j}}$ where the coefficients $c_{m,j}$ can be pseudorandom numbers governed by S. When S is known, all the polynomials are known and yet $\mathbf{x}$ is still generally hard to obtain from $\mathbf{y}$ for any M due to the nonlinearity. But we can write $y_m = g_m(\mathbf{v}(x_1, \cdots, x_N))$, where g_m is a scalar linear function conditioned on S, and $\mathbf{v}(x_1, \cdots, x_N)$ is a $J \times 1$ vector nonlinear function unconditioned on S. This means that the HOP is not a hard-to-substitute function. It is also obvious that HOP is generally highly sensitive to noise in $\mathbf{x}$ due to higher-order polynomials. Specifically, $\partial y_m = p_{1,j}\sum_{j=0}^{J} c_{m,j}(x_1^{p_{1,j}-1}x_2^{p_{2,j}}\cdots x_N^{p_{N,j}})\partial x_1 + \cdots + p_{N,j}\sum_{j=0}^{J} c_{m,j}(x_1^{p_{1,j}}\cdots x_{N-1}^{p_{N-1,j}}x_N^{p_{N,j}-1})\partial x_N$, where ∂ denotes the differential operator. A large $p_{i,j}$ means a large sensitivity to noise in x_i. So, HOP does not seem a good choice in applications. It is obvious that the per-sample complexity order of the HOP is $\mathcal{O}(\sum_{j=0}^{J}\sum_{i=1}^{N} p_{i,j})$, a simpler form of which in terms of a large N is $\mathcal{O}(N)$.

3.2. Index-of-max hashing

More recently a method called index-of-max (IoM) hashing was proposed in [11] and applied in [13]. There are algorithms 1 and 2 based on IoM, which will be referred to as IoM-1 and IoM-2.

In IoM-1, the feature vector $\mathbf{x} \in \mathcal{R}^N$ is multiplied (from the left) by a sequence of $L \times N$ pseudorandom matrices $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ to produce $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$ respectively. The index of the largest element in each $\mathbf{v}_k$ is used as an output y_k. With $\mathbf{y} = [y_1, \cdots, y_{K_1}]^T$, we see that $\mathbf{y}$ is a nonlinear ("piece-wise" constant and "piece-wise" continuous) continuous function of $\mathbf{x}$.

The generation of each of $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ requires $\mathcal{O}(N^2)$ complexity, and the computation of each of $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$ requires additional $\mathcal{O}(N^2)$ complexity. The search for the maximum entry within each $\mathbf{v}_k$ costs $\mathcal{O}(N)$. Hence, the per-sample complexity of IoM-1 is $\mathcal{O}(N^2)$.

Table 2
Normalized projection of $\mathbf{x}$ onto its estimate using only averaging for attack of IoM-1.

	$K_1 = 8$	16	32	64
$N = 8$	0.8546	0.9171	0.9562	0.9772
16	0.8022	0.8842	0.9365	0.9666
32	0.7328	0.8351	0.906	0.9494

In IoM-2, $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ used in IoM-1 are replaced by $N \times N$ pseudorandom permutation matrices $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$ to produce $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$, and then a sequence of vectors $\mathbf{h}_1, \cdots, \mathbf{h}_{K_2}$ are produced in such a way that each $\mathbf{h}_k$ is the element-wise products of an exclusive set of p vectors from $\mathbf{v}_1, \cdots, \mathbf{v}_{K_1}$. The index of the largest element in each $\mathbf{h}_k$ is used as an output y_k. With $\mathbf{y} = [y_1, \cdots, y_{K_2}]^T$, we see that $\mathbf{y}$ is another nonlinear continuous function of $\mathbf{x}$.

The complexity of p random permutations of $\mathbf{x}$ to produce p of $\mathbf{v}_k$ is $\mathcal{O}(pN^2)$ (even though there is no multiplication required). The complexity to produce each $\mathbf{h}_k$ is $\mathcal{O}(pN)$. Then the per-sample complexity of IoM-2 is also $\mathcal{O}(N^2)$ provided that p is independent of N. If $p = N$, the per-sample complexity of IoM-2 becomes $\mathcal{O}(N^3)$.

Next we show that IoM-1 is not hard to invert if the secret key S or equivalently the random matrices $\mathbf{R}_1, \cdots, \mathbf{R}_{K_1}$ are known. We also show that IoM-2 is not hard to invert up to the sign of each element in $\mathbf{x}$ if the secret key S or equivalently the random permutations $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$ are known.

3.2.1. Attack of IoM-1

Assume that each $\mathbf{R}_k$ has L rows and the secret key S is known. Then knowing y_k for $k = 1, \cdots, K_1$ means knowing $\mathbf{r}_{k,a,l}$ and $\mathbf{r}_{k,b,l}$ satisfying

$$\mathbf{r}_{k,a,l}^T\mathbf{x} > \mathbf{r}_{k,b,l}^T\mathbf{x} \tag{41}$$

with $l = 1, \cdots, L-1$ and $k = 1, \cdots, K_1$. Here $\mathbf{r}_{k,a,l}^T$ and $\mathbf{r}_{k,b,l}^T$ for all l are rows of $\mathbf{R}_k$. The above is equivalent to $\mathbf{d}_{k,l}^T\mathbf{x} > 0$ with $\mathbf{d}_{k,l} = \mathbf{r}_{k,a,l} - \mathbf{r}_{k,b,l}$, or more simply

$$\mathbf{d}_k^T\mathbf{x} > 0 \tag{42}$$

where $\mathbf{d}_k$ is known for $k = 1, \cdots, K$ with $K = K_1(L-1)$. Note that any scalar change to $\mathbf{x}$ does not affect the output $\mathbf{y}$. Also note that even though IoM-1 defines a nonlinear function from $\mathbf{x}$ to $\mathbf{y}$, the conditions in (42) useful for attack are linear with respect to $\mathbf{x}$.

To attack IoM-1, we can simply compute $\hat{\mathbf{x}}$ satisfying $\mathbf{d}_k^T\hat{\mathbf{x}} > 0$ for all k. One such algorithm of attack is as follows:

1. Initialization/averaging: Let $\hat{\mathbf{x}} = \bar{\mathbf{d}} \doteq \frac{1}{K}\sum_{k=1}^{K}\mathbf{d}_k$.
2. Refinement: Until $\mathbf{d}_k^T\hat{\mathbf{x}} > 0$ for all k, choose $k^* = arg\min_k \mathbf{d}_k^T\hat{\mathbf{x}}$, and compute

$$\hat{\mathbf{x}} \leftarrow \hat{\mathbf{x}} - \eta(\mathbf{d}_{k^*}^T\hat{\mathbf{x}})\mathbf{d}_{k^*} \tag{43}$$

where η is a step size.

Our simulation (using $\eta = \frac{1}{\|\mathbf{d}_{k^*}\|^2}$) shows that using the initialization alone can yield a good estimate of $\mathbf{x}$ as K increases. More specifically, the normalized projection $\frac{\bar{\mathbf{d}}^T\mathbf{x}}{\|\bar{\mathbf{d}}\|\cdot\|\mathbf{x}\|}$ converges to one as K increases. Our simulation also shows that the second step in the above algorithm improves the convergence slightly. Examples of the attack results are shown in Tables 2 and 3 where $L = N$. We see that IoM-1 (with its key S exposed) can be inverted with a complexity order no larger than a linear function of N and K_1 respectively.

3.2.2. Attack of IoM-2

To attack IoM-2, we need to know the sign of each element of $\mathbf{x}$, which is assumed below. Given the output of IoM-2 and all the

Table 3
Normalized projection of $\mathbf{x}$ onto its estimate after convergence of refinement for attack of IoM-1.

	$K_1 = 8$	16	32	64
$N = 8$	0.8807	0.9467	0.9804	0.9937
16	0.8174	0.908	0.9612	0.9861
32	0.739	0.8497	0.9268	0.9699

Table 4
Normalized projection of $|\mathbf{x}|$ onto its estimate using only averaging for attack of IoM-2.

	$K_2 = 8$	16	32	64
$N = 8$	0.9244	0.954	0.9698	0.9783
16	0.9068	0.9418	0.9603	0.9694
32	0.8844	0.9206	0.9379	0.9466

Table 5
Normalized projection of $|\mathbf{x}|$ onto its estimate after convergence of refinement for attack of IoM-2.

	$K_2 = 8$	16	32	64
$N = 8$	0.9432	0.9711	0.9802	0.9816
16	0.9182	0.9525	0.9649	0.9653
32	0.8887	0.9258	0.9403	0.9432

permutation matrices $\mathbf{P}_1, \cdots, \mathbf{P}_{K_1}$, we know which of the elements in each $\mathbf{h}_k$ is the largest and which of these elements are negative. If the largest element in $\mathbf{h}_k$ is positive, we will ignore all the negative elements in $\mathbf{h}_k$. If the largest element in $\mathbf{h}_k$ is negative, we know which of the elements in $\mathbf{h}_k$ has the smallest absolute value.

Let $|\mathbf{h}_k|$ be the vector consisting of the corresponding absolute values of the elements in $\mathbf{h}_k$. Also let $\log|\mathbf{h}_k|$ be the vector of element-wise logarithm of $|\mathbf{h}_k|$. It follows that

$$\log|\mathbf{h}_k| = \mathbf{T}_k \log|\mathbf{x}| \tag{44}$$

where $\mathbf{T}_k$ is the sum of the permutation matrices used for $\mathbf{h}_k$. The knowledge of an output y_k of IoM-2 implies the knowledge of $\mathbf{t}_{k,a,l}^T$ and $\mathbf{t}_{k,b,l}^T$ (i.e., row vectors of $\mathbf{T}_k$) such that either

$$\mathbf{t}_{k,a,l}^T \log|\mathbf{x}| > \mathbf{t}_{k,b,l} \log|\mathbf{x}| \tag{45}$$

with $l = 1, \cdots, L_k - 1$ if $\mathbf{h}_k$ has $L_k \geq 2$ positive elements, or

$$\mathbf{t}_{k,a,l}^T \log|\mathbf{x}| < \mathbf{t}_{k,b,l} \log|\mathbf{x}| \tag{46}$$

with $l = 1, \cdots, N - 1$ if $\mathbf{h}_k$ has no positive element.

If $\mathbf{h}_k$ has only one positive element, the corresponding y_k is ignored as it yields no useful constraint on $\log|\mathbf{x}|$. We assume that no element in $\mathbf{x}$ is zero.

Equivalently, the knowledge of y_k implies $\mathbf{c}_{k,l}^T \log|\mathbf{x}| > 0$ where $\mathbf{c}_{k,l} = \mathbf{t}_{k,a,l} - \mathbf{t}_{k,b,l}$ for $l = 1, \cdots, L_k - 1$ if $\mathbf{h}_k$ has $L_k \geq 2$ positive elements, or $\mathbf{c}_{k,l} = -\mathbf{t}_{k,a,l} + \mathbf{t}_{k,b,l}$ for $l = 1, \cdots, N - 1$ if $\mathbf{h}_k$ has no positive element. A simpler form of the constraints on $\log|\mathbf{x}|$ is

$$\mathbf{c}_k^T \log|\mathbf{x}| > 0 \tag{47}$$

where $\mathbf{c}_k$ is known for $k = 1, \cdots, K$ with $K = \sum_{k=1}^{K_2}(\bar{L}_k - 1)$. Here $\bar{L}_k = L_k$ if $\mathbf{h}_k$ has a positive element, and $\bar{L}_k = N$ if $\mathbf{h}_k$ has no positive element.

The algorithm to find $\log|\mathbf{x}|$ satisfying (47) for all k is similar to that for (42), which consists of "initialization/averaging" and "refinement". Knowing $\log|\mathbf{x}|$, we also know $|\mathbf{x}|$. Examples of the attack results are shown in Tables 4 and 5 where $p = N$ and all entries of $\mathbf{x}$ are assumed to be positive.

The above analysis shows that IoM-2 effectively extracts out a binary (sign) secret from each element of $\mathbf{x}$ and utilizes that secret to construct its output. Other than that secret, IoM-2 is not a hard-to-invert function. In other words, IoM-2 can be inverted with a complexity order no larger than $L_{N,K_2}2^N$ where L_{N,K_2} is a linear function of N and K_2, respectively, and 2^N is to due to an exhaustive search of the sign of each element in $\mathbf{x}$. Note that if an additional key S_x of N bits is first extracted with 100% reliability from the signs of the elements in $\mathbf{x}$, then a linear CEF could be used while maintaining an attack complexity order equal to $\mathcal{O}(N^3 2^N)$.

4. A new family of nonlinear CEFs

The previous discussions show that RP, DRP and IoM-1 are not hard to invert, and IoM-2 can be inverted with a complexity order no larger than $L_{N,K_2}2^N$. We show next a new family of nonlinear CEFs, for which the best known method to attack suffers a complexity order no less than $\mathcal{O}(2^{\zeta N})$ with ζ substantially larger than one.

The new family of nonlinear CEFs is broadly defined as follows. Step 1: let $\mathbf{M}_{k,x}$ be a matrix (for index k) consisting of elements that result from a random modulation of the input vector $\mathbf{x} \in \mathcal{R}^N$. Step 2: Each element of the output vector $\mathbf{y} \in \mathcal{R}^M$ is constructed from a component of the singular value decomposition (SVD) of $\mathbf{M}_{k,x}$ for some k. Each of the two steps can have many possibilities. We will next focus on one specific CEF in this family (as this CEF seems the best among many choices we have considered).

For each pair of k and l, let $\mathbf{Q}_{k,l}$ be a (secret key dependent) random $N \times N$ unitary (real) matrix. Define

$$\mathbf{M}_{k,x} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}] \tag{48}$$

where each column of $\mathbf{M}_{k,x}$ is a random rotation of $\mathbf{x}$. Let $\mathbf{u}_{k,x,1}$ be the principal left singular vector of $\mathbf{M}_{k,x}$, i.e.,

$$\mathbf{u}_{k,x,1} = arg \max_{\mathbf{u}, \|\mathbf{u}\|=1} \mathbf{u}^T \mathbf{M}_{k,x} \mathbf{M}_{k,x}^T \mathbf{u} \tag{49}$$

Then for each k, choose N_y $(1 \leq N_y < N)$ elements in $\mathbf{u}_{k,x,1}$ to be N_y elements in $\mathbf{y} = [y_1, y_2, \cdots]^T$. If we choose $N_y = 1$, then y_k for each k is an entry (such as the 1st entry) of $\mathbf{u}_{k,x,1}$. We will refer to the above function (from $\mathbf{x}$ to $\mathbf{y}$) as SVD-CEF. Note that there are efficient ways to perform the forward computation needed for (49) given $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$. One of them is the power method [19], which has the complexity equal to $\mathcal{O}(N^2)$. But the construction of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$ (starting from the generation of $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}$) for each k requires $\mathcal{O}(N^3)$ complexity.

We can see that for each random realization of $\mathbf{Q}_{k,l}$ for all k and l and a random realization $\mathbf{x}_0$ of $\mathbf{x}$, with probability one there is a neighborhood around $\mathbf{x}_0$ within which $\mathbf{y}$ is a continuous function of $\mathbf{x}$. It is also clear that for any fixed $\mathbf{x}$ the elements in $\mathbf{y}$ appear random to anyone who does not have access to the secret key used to produce the pseudorandom $\mathbf{Q}_{k,l}$.

More importantly, we will show in Section 5 that SVD-CEF is empirically hard to attack even with $\mathbf{Q}_{k,l}$ known for all k and l; and in Section 6 that if $\mathbf{x}$ consists of i.i.d. $\mathcal{N}(0, \sigma_x^2)$, then all entries of $\mathbf{y} = [y_1, y_2, \cdots]^T$ have nearly zero correlations and the same distribution even with $\mathbf{Q}_{k,l}$ being fixed for all k and l. The noise sensitivity of SVD-CEF is also discussed in Section 6.

5. Attack of SVD-CEF

We now consider how to compute $\mathbf{x} \in \mathcal{R}^N$ from a given $\mathbf{y} \in \mathcal{R}^M$ with $M \geq N$ for SVD-CEF based on (48) and (49) assuming that $\mathbf{Q}_{k,l}$ for all k and l are given.

A universal method for inverting a function is via exhaustive search, i.e., searching for a $\mathbf{x}$ that produces the known $\mathbf{y}$ via the forward function up to a desired precision. This method has a complexity order no less than $\mathcal{O}(2^{N_B N})$ with N_B being an effective number of bits needed to represent each of the N elements in $\mathbf{x}$. The value of N_B depends on an expected noise level in $\mathbf{x}$. It is not uncommon in practice that N_B ranges from 3 to 8 or even higher.

The only other known method that we know to invert SVD-CEF is the Newton's method, which is considered next. To prepare for the application of the Newton's method, we need to formulate a set of equations which must be satisfied by all unknown variables.

5.1. Preparation

We now assume that for each of $k = 1, \cdots, K$, N_y elements of $\mathbf{u}_{k,x,1}$ are used to construct $\mathbf{y} \in \mathcal{R}^M$ with $M = KN_y$. Computing $\mathbf{x}$ from $\mathbf{y}$ and $\mathbf{Q}_{k,l}$ for all k and l is equivalent to solving the following eigenvalue-decomposition (EVD) equations:

$$\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T\mathbf{u}_{k,x,1} = \sigma_{k,x,1}^2\mathbf{u}_{k,x,1} \tag{50}$$

with $k = 1, \cdots, K$. Here $\sigma_{k,x,1}^2$ is the principal eigenvalue of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$. But this is not a conventional EVD problem because the vector $\mathbf{x}$ inside $\mathbf{M}_{k,x}$ is unknown along with $\sigma_{k,x,1}^2$ and $N - N_y$ elements in $\mathbf{u}_{k,x,1}$ for each k. We will refer to (50) as the EVD equilibrium conditions for $\mathbf{x}$.

If the unknown $\mathbf{x}$ is multiplied by α, so should be the corresponding unknowns $\sigma_{k,x,1}$ for all k but $\mathbf{u}_{k,x,1}$ for any k is not affected. So, we will only need to consider the solution satisfying $\|\mathbf{x}\|^2 = 1$. Note that if the norm of the original feature vector contains secret, we can first use the transformation shown in Section 2.3.1.

The number of unknowns in the system of nonlinear equations (50) is $N_{unk,EVD,1} = N + (N - N_y)K + K$, which consists of all N elements of $\mathbf{x}$, $N - N_y$ elements of $\mathbf{u}_{k,x,1}$ for each k and $\sigma_{k,x,1}^2$ for all k. The number of the nonlinear equations is $N_{equ,EVD,1} = NK + K + 1$, which consists of (50) for all k, $\|\mathbf{u}_{k,x,1}\| = 1$ for all k and $\|\mathbf{x}\|^2 = 1$. Then, the necessary condition for a finite set of solutions is $N_{equ,EVD,1} \geq N_{unk,EVD,1}$, or equivalently $N_yK \geq N - 1$.

If $N_y < N$, there are $N - N_y$ unknowns in $\mathbf{u}_{k,x,1}$ for each k and hence the left side of (50) is a third-order function of unknowns. To reduce the nonlinearity, we can expand the space of unknowns as follows. Since $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T = \sum_{l=1}^N \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T$ with $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ (a substitute input), we can treat $\mathbf{X}$ as a $N \times N$ symmetric unknown matrix (without the rank-1 constraint), and rewrite (50) as

$$(\sum_{l=1}^N \mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T)\mathbf{u}_{k,x,1} = \sigma_{k,x,1}^2\mathbf{u}_{k,x,1} \tag{51}$$

with $Tr(\mathbf{X}) = 1$, $\|\mathbf{u}_{k,x,1}\| = 1$ and $k = 1, \cdots, K$. In this case, both sides of (51) are of the 2nd order of all unknowns. But the number of unknowns is now $N_{unk,EVD,2} = \frac{1}{2}N(N+1) + (N - N_y)K + K > N_{unk,EVD,1}$ while the number of equations is not changed, i.e., $N_{equ,EVD,2} = N_{equ,EVD,1} = NK + K + 1$. In this case, the necessary condition for a finite set of solution for $\mathbf{X}$ is $N_{equ,EVD,2} \geq N_{unk,EVD,2}$, or equivalently $N_yK \geq \frac{1}{2}N(N+1) - 1$.

Note that $\mathbf{X}$ seems the only useful substitute for $\mathbf{x}$. But this substitute still seems hard to compute from $\mathbf{y}$ as shown later.

Alternatively, we know that $\mathbf{x}$ satisfies the following SVD equations:

$$\mathbf{M}_{k,x}\mathbf{V}_{k,x} = \mathbf{U}_{k,x}\mathbf{\Sigma}_{k,x} \tag{52}$$

with $\mathbf{U}_{k,x}^T\mathbf{U}_{k,x} = \mathbf{I}_N$ and $\mathbf{V}_{k,x}^T\mathbf{V}_{k,x} = \mathbf{I}_N$. Here $\mathbf{U}_{k,x}$ is the matrix of all left singular vectors, $\mathbf{V}_{k,x}$ is the matrix of all right singular vectors, and $\mathbf{\Sigma}_{k,x}$ is the diagonal matrix of all singular values. The above equations are referred to as the SVD equilibrium conditions on $\mathbf{x}$.

With N_y elements of the first column of $\mathbf{U}_{k,x}$ for each k to be known, the unknowns are the vector $\mathbf{x}$, $N^2 - N_y$ elements in $\mathbf{U}_{k,x}$ for each k, all N^2 elements in $\mathbf{V}_{k,x}$ for each k, and all diagonal elements in $\mathbf{\Sigma}_{k,x}$ for each k. Then, the number of unknowns is now $N_{unk,SVD} = N + (N^2 - N_y)K + N^2K + NK$, and the number of equations is $N_{equ,SVD} = N^2K + N(N+1)K + 1$. In this case, $N_{equ,SVD} \geq N_{unk,SVD}$ iff $N_yK \geq N - 1$. This is the same condition as that for EVD equilibrium. But the SVD equilibrium equations in (52) are all of the second order.

Note that for the EVD equilibrium, there is no coupling between different eigen-components. But for the SVD equilibrium, there are couplings among all singular-components. Hence the latter involves a much larger number of unknowns than the former. Specifically, $N_{unk,SVD} > N_{unk,EVD,2} > N_{unk,EVD,1}$.

Every set of equations that $\mathbf{x}$ must fully satisfy (given $\mathbf{y}$) is a set of nonlinear equations, regardless of how the parameterization is chosen. This seems the fundamental reason why SVD-CEF is hard to invert. SVD is a three-factor decomposition of a real-valued matrix, for which there are efficient ways for forward computations but no easy way for backward computation. If a two-factor decomposition of a real-valued matrix (such as QR decomposition) is used, the hard-to-invert property does not seem achievable.

In Appendix A, the details of an attack algorithm based on Newton's method are given.

5.2. Performance of attack algorithm

Since the conditions useful for attack of SVD-CEF are always nonlinear, any attack algorithm with a random initialization $\mathbf{x}'$ can converge to the true vector $\mathbf{x}$ (or its equivalent which produces the same $\mathbf{y}$) only if $\mathbf{x}'$ is close enough to $\mathbf{x}$. To translate the local convergence into a computational complexity needed to successfully obtain $\mathbf{x}$ from $\mathbf{y}$, we now consider the following.

Let $\mathbf{x}$ be an N-dimensional unit-norm vector of interest. Any unit-norm initialization of $\mathbf{x}$ can be written as

$$\mathbf{x}' = \pm\sqrt{1 - r^2}\mathbf{x} + r\mathbf{w} \tag{53}$$

where $0 < r \leq 1$ and $\mathbf{w}$ is a unit-norm vector orthogonal to $\mathbf{x}$. For any $\mathbf{x}$, $r\mathbf{w}$ is a vector (or "point") on the sphere of dimension $N - 2$ and radius r, denoted by $\mathcal{S}^{N-2}(r)$. The total area of $\mathcal{S}^{N-2}(r)$ is known to be $|\mathcal{S}^{N-2}(r)| = \frac{2\pi^{\frac{N-1}{2}}}{\Gamma(\frac{N-1}{2})}r^{N-2}$. Then the probability for a uniformly random $\mathbf{x}'$ from $\mathcal{S}^{N-1}(1)$ to fall onto $\mathcal{S}^{N-2}(r_0)$ orthogonal to $\sqrt{1 - r_0^2}\mathbf{x}$ with $r \leq r_0 \leq r + dr$ is $2\frac{|\mathcal{S}^{N-2}(r)|}{|\mathcal{S}^{N-1}(1)|}dr$ where the factor 2 accounts for $\pm$ in (53).

Therefore, the probability of convergence from $\mathbf{x}'$ to $\mathbf{x}$ is

$$\begin{aligned} P_{conv} &= \mathcal{E}_x\left\{\int_0^1 2P_{x,r}\frac{|\mathcal{S}^{N-2}(r)|}{|\mathcal{S}^{N-1}(1)|}dr\right\} \\ &= \frac{2\Gamma\left(\frac{N}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{N-1}{2}\right)}\int_0^1 P_r r^{N-2}dr \end{aligned} \tag{54}$$

where $\mathcal{E}_x$ is the expectation over $\mathbf{x}$, $P_{x,r}$ is the probability of convergence from $\mathbf{x}'$ to $\mathbf{x}$ when $\mathbf{x}'$ is chosen randomly from $\mathcal{S}^{N-2}(r)$ orthogonal to a given $\sqrt{1 - r^2}\mathbf{x}$, and $\mathcal{E}_x\{P_{x,r}\} = P_r$.

We see that P_r is the probability that the algorithm converges from $\mathbf{x}'$ to $\mathbf{x}$ (including its equivalent) subject to a fixed r, uniformly random unit-norm $\mathbf{x}$, and uniformly random unit-norm $\mathbf{w}$ satisfying $\mathbf{w}^T\mathbf{x} = 0$. And P_r can be estimated via simulation.

Let $r_{\max} < 1$ be such that $P_r = 0$ for $r \geq r_{\max}$. Then

$$\begin{aligned} P_{conv} &= \frac{2\Gamma\left(\frac{N}{2}\right)}{\sqrt{\pi}\Gamma\left(\frac{N-1}{2}\right)}\int_0^{r_{max}} P_r r^{N-2}dr \\ &< \frac{2\Gamma\left(\frac{N}{2}\right)}{(N-1)\sqrt{\pi}\Gamma\left(\frac{N-1}{2}\right)}r_{max}^{N-1} \\ &< r_{max}^{N-1} \end{aligned} \tag{55}$$

which converges to zero exponentially as N increases. In other words, for such an algorithm to find $\mathbf{x}$ or its equivalent from random initializations has a complexity order equal to $\mathcal{O}(\frac{1}{P_{conv}}) > \mathcal{O}((\frac{1}{r_{\max}})^{N-1})$ which increases exponentially as N increases.

Table 6
$P_{r,N}$ and $P^*_{r,N}$ versus r and N.

r	0.001	0.01	0.1	0.3	0.5	0.7	0.9	1
$P_{r,4}$	0.46	0.24	0.06	0	0.01	0.01	0.01	0
$P^*_{r,4}$	0.45	0.17	0.04	0	0.01	0	0.01	0
$P_{r,8}$	0.29	0.07	0.01	0	0	0	0	0
$P^*_{r,8}$	0.25	0.05	0	0	0	0	0	0

In our simulation, we have found that $r_{\max}$ decreases rapidly as N increases. Let $P_{r,N}$ be P_r as function of N. Also let $P^*_{r,N}$ be the probability of convergence to $\hat{\mathbf{x}}$ which via SVD-CEF not only yields the correct y_k for $k=1,\cdots,K$ but also the correct y_k for $k>K$ (up to maximum absolute element-wise error no larger than 0.02). Here K is the number of output elements used to compute the input vector $\mathbf{x}$. In the simulation, we chose $N_y=1$ and $N_{equ,EVD,2}=N_{unk,EVD,2}+1$, which is equivalent to $K=\frac{1}{2}N(N+1)$. Shown in Table 6 are the percentage values of $P_{r,N}$ versus r and N, which are based on 100 random choices of $\mathbf{x}$. For each choice of $\mathbf{x}$ and each value of r, we used one random initialization of $\mathbf{x}'$. (For $N=8$ and the values of r in this table, it took two days on a PC with CPU 3.4 GHz Dual Core to complete the 100 runs.)

The above discussions have explained why SVD-CEF is empirically hard to attack. Next we will discuss the sensitivity, correlation and invariance of SVD-CEF.

6. Statistics of SVD-CEF

In this section, we show a statistical study of SVD-CEF to understand some of the statistical properties of its output. Since each entry of the output $\mathbf{y}=[y_1,y_2,\cdots,y_M]^T$ of SVD-CEF is an element in the principal eigenvector $\mathbf{u}_{k,x,1}$ of the matrix $\mathbf{M}_{k,x}\mathbf{M}^T_{k,x}$, we can mostly focus on the statistics of $\mathbf{u}_{k,x,1}$.

6.1. Sensitivity

Unlike the unitary random projections, here the relationship between the normalized distance at the input $\frac{1}{\sqrt{N}}\|\Delta\mathbf{x}\|$ and the normalized distance at the output $\frac{1}{\sqrt{M}}\|\Delta\mathbf{y}\|$ is not trivial.

6.1.1. Sensitivity to small perturbation

We now consider the sensitivity of SVD-CEF to a small perturbation, i.e., the relationship between the differential $\partial\mathbf{u}_{k,x,1}$ (or a corresponding ∂y_k) and the differential $\partial\mathbf{x}$. It follows from [21] that

$$\partial\mathbf{u}_{k,x,1}=\sum_{j=2}^{N}\frac{1}{\lambda_1-\lambda_j}\mathbf{u}_{k,x,j}\mathbf{u}^T_{k,x,j}\partial(\mathbf{M}_{k,x}\mathbf{M}^T_{k,x})\mathbf{u}_{k,x,1}. \tag{56}$$

where λ_j is the jth eigenvalue of $\mathbf{M}_{k,x}\mathbf{M}^T_{k,x}$, and $\mathbf{u}_{k,x,j}$ is the corresponding jth eigenvector. Since $\mathbf{M}_{k,x}\mathbf{M}^T_{k,x}=\sum_{l=1}^{N}\mathbf{Q}_{k,l}\mathbf{x}\mathbf{x}^T\mathbf{Q}^T_{k,l}$, $\partial(\mathbf{M}_{k,x}\mathbf{M}^T_{k,x})=\sum_l\mathbf{Q}_{k,l}\partial\mathbf{x}\mathbf{x}^T\mathbf{Q}^T_{k,l}+\sum_l\mathbf{Q}_{k,l}\mathbf{x}\partial\mathbf{x}^T\mathbf{Q}^T_{k,l}$. It follows that

$$\partial\mathbf{u}_{k,x,1}=\mathbf{T}\partial\mathbf{x} \tag{57}$$

where $\mathbf{T}=\mathbf{A}+\mathbf{B}$ with

$$\mathbf{A}=\sum_{j=2}^{N}\frac{1}{\lambda_1-\lambda_j}\mathbf{u}_{k,x,j}\mathbf{u}^T_{k,x,j}\sum_{l=1}^{N}\mathbf{Q}_{k,l}\mathbf{x}^T\mathbf{Q}^T_{k,l}\mathbf{u}_{k,x,1} \tag{58}$$

$$\mathbf{B}=\sum_{j=2}^{N}\frac{1}{\lambda_1-\lambda_j}\mathbf{u}_{k,x,j}\mathbf{u}^T_{k,x,j}\sum_{l=1}^{N}\mathbf{Q}_{k,l}\mathbf{x}\mathbf{u}^T_{k,x,1}\mathbf{Q}_{k,l}. \tag{59}$$

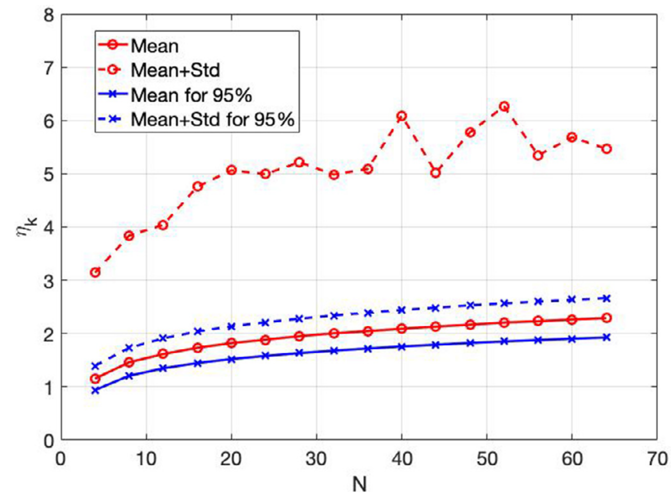

Fig. 1. The mean and mean-plus-deviation of $\eta_{k,x}$ versus N.

We can also write

$$\mathbf{T}=\left(\sum_{j=2}^{N}\frac{1}{\lambda_1-\lambda_j}\mathbf{u}_{k,x,j}\mathbf{u}^T_{k,x,j}\right)\cdot\left(\sum_{l=1}^{N}\mathbf{Q}_{k,l}\left[(\mathbf{x}^T\mathbf{Q}^T_{k,l}\mathbf{u}_{k,x,1})\mathbf{I}_N+\mathbf{x}\mathbf{u}^T_{k,x,1}\mathbf{Q}_{k,l}\right]\right) \tag{60}$$

where the first matrix component has the rank $N-1$ and hence so does $\mathbf{T}$.

Let $\partial\mathbf{x}=\mathbf{w}$ which consists of i.i.d. elements with zero mean and variance $\sigma_w^2\ll 1$. It then follows that

$$\mathcal{E}_w\{\|\partial\mathbf{u}_{k,x,1}\|^2\}=Tr\{\mathbf{T}\sigma_w^2\mathbf{T}^T\}=\sigma_w^2\sum_{j=1}^{N-1}\sigma_j^2 \tag{61}$$

where σ_j for $j=1,\cdots,N-1$ are the nonzero singular values of $\mathbf{T}$. Since $\mathcal{E}_w\{\|\partial\mathbf{x}\|^2\}=N\sigma_w^2$, we have

$$\eta_{k,x}=\sqrt{\frac{\mathcal{E}_w\{\|\partial\mathbf{u}_{k,x,1}\|^2\}}{\mathcal{E}_w\{\|\partial\mathbf{x}\|^2\}}}=\sqrt{\frac{1}{N}\sum_{j=1}^{N-1}\sigma_j^2} \tag{62}$$

which measures the sensitivity of $\mathbf{u}_{k,x,1}$ to a small perturbation in $\mathbf{x}$.

Since each of the N entries in $\partial\mathbf{u}_{k,x,1}$ has the same variance due to symmetry, then the corresponding ∂y_k satisfies $\mathcal{E}_w\{\|\partial y_k\|^2\}=\frac{1}{N}\mathcal{E}_w\{\|\partial\mathbf{u}_{k,x,1}\|^2\}$. Since both $\mathbf{x}$ and $\mathbf{u}_{k,x,1}$ have the unit norm, the input SNR of SVD-CEF is $\mathtt{SNR}_x=1/\mathcal{E}_w\{\|\partial\mathbf{x}\|^2\}=\frac{1}{N\sigma_w^2}$, and the output SNR of SVD-CEF for y_k is $\mathtt{SNR}_{y,k}=\mathcal{O}(\frac{1}{N\mathcal{E}_w\{\|\partial y_k\|^2\}})=\mathcal{O}(1/\mathcal{E}_w\{\|\partial\mathbf{u}_{k,x,1}\|^2\})$. Therefore, the FoM of SVD-CEF for y_k is $\sqrt{\frac{\mathtt{SNR}_x}{\mathtt{SNR}_{y,k}}}=\mathcal{O}(\eta_{k,x})$. Here $\mathcal{O}$ denotes the order as $\sigma_w^2\to 0$.

For each given $\mathbf{x}$, there is a small percentage of realizations of $\{\mathbf{Q}_{k,l},l=1,\cdots,N\}$ that make $\eta_{k,x}$ relatively large. To reduce $\eta_{k,x}$, we can prune away such bad realizations.

Shown in Fig. 1 are the means and means-plus-deviations of $\eta_{k,x}$ (over choices of k and $\mathbf{x}$) versus N, with and without pruning respectively. Here "std" stands for standard deviation. We see that 5% pruning (or equivalently 95% inclusion shown in the figure) results in a substantial reduction of $\eta_{k,x}$. We used 1000×1000 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,l},l=1,\cdots,N\}$. Shown in Table 7 are statistics of $\eta_{k,x}$ subject to $\eta_{k,x}<2.5$ where P_{good} is the probability of $\eta_{k,x}<2.5$. We see that P_{good} is relatively large at around or above 80% and the mean of $\eta_{k,x}$ ranges roughly from 1.3 to 1.6 for $N=16,32,64$. This noise sensitivity is far from perfect when compared

Table 7
Statistics of $\eta_{k,x}$ subject to $\eta_{k,x} < 2.5$ and P_{good}.

N	16	32	64
Mean	1.325	1.489	1.645
Std	0.414	0.397	0.371
P_{good}	0.88	0.84	0.78

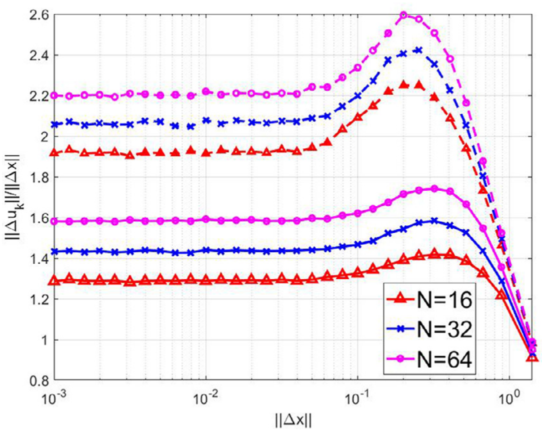

Fig. 2. The means (lower three curves) and means-plus-deviations (upper three curves) of $\frac{\|\Delta \mathbf{u}_{k,x,1}\|}{\|\Delta \mathbf{x}\|}$ subject to $\eta_{k,x} < 2.5$.

to the unitary random projection. But SVD-CEF has the hard-to-attack property as empirically established earlier.

6.1.2. Sensitivity to large perturbation

Any unit-norm vector $\mathbf{x}'$ can be written as $\mathbf{x}' = \pm\sqrt{1-\alpha}\mathbf{x} + \sqrt{\alpha}\mathbf{w}$ where $0 \le \alpha \le 1$, and $\mathbf{w}$ is of the unit norm and satisfies $\mathbf{w}^T\mathbf{x} = 0$. Then $\|\Delta\mathbf{x}\| = \|\mathbf{x}' - \mathbf{x}\| = \sqrt{2 - 2\sqrt{1-\alpha}}$. It follows that $\|\Delta\mathbf{x}\| \le \sqrt{2}$ and $\|\Delta\mathbf{u}_{k,x,1}\| \le \sqrt{2}$. For given α in $\mathbf{x}' = \pm\sqrt{1-\alpha}\mathbf{x} + \sqrt{\alpha}\mathbf{w}$, $\|\Delta\mathbf{x}\|$ is given while $\|\Delta\mathbf{u}_{k,x,1}\|$ still depends on $\mathbf{w}$. We can call $\frac{\|\Delta\mathbf{u}_{k,x,1}\|}{\|\Delta\mathbf{x}\|}$ a deviation gain of SVD-CEF, which is dependent on $\mathbf{x}$, k and $\|\Delta\mathbf{x}\|$. Here a different k means a different set of $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$. Shown in Fig. 2 are the means and means-plus-deviations of the deviation gain versus $\|\Delta\mathbf{x}\|$ subject to $\eta_{k,x} < 2.5$. This figure is based on 1000×1000 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,l}, l = 1, \cdots, N\}$. We see that the mean of the deviation gain is somewhat constant and comparable to the mean of $\eta_{k,x}$ for $\|\Delta\mathbf{x}\| < 0.1$.

6.2. Correlation

We show below via simulation that the correlation between the input and output of SVD-CEF as well as the correlation among the output samples of SVD-CEF are practically zero.

6.2.1. Correlation between input and output

Recall $\mathbf{M}_{k,x} = [\mathbf{Q}_{k,1}\mathbf{x}, \cdots, \mathbf{Q}_{k,N}\mathbf{x}]$. If there is a secret key, then $\mathbf{Q}_{k,l}$ for all k and l are uniformly random unitary matrices (from adversary's perspective). Then $\mathbf{u}_{k,x,1}$ for all k and any $\mathbf{x}$ are uniformly random on $\mathcal{S}^{N-1}(1)$. It follows that $\mathcal{E}_Q\{\mathbf{u}_{k,x,1}\mathbf{u}_{m,x,1}^T\} = 0$ for $k \neq m$, and $\mathcal{E}_Q\{\mathbf{u}_{k,x,1}\mathbf{x}^T\} = 0$. Furthermore, it can be shown that $\mathcal{E}_Q\{\mathbf{u}_{k,x,1}\mathbf{u}_{k,x,1}^T\} = \frac{1}{N}\mathbf{I}_N$, i.e., the entries of $\mathbf{u}_{k,x,1}$ are uncorrelated with each other. Here $\mathcal{E}_Q$ denotes the expectation over the distributions of $\mathbf{Q}_{k,l}$.

If there is no secret key, then $\mathbf{Q}_{k,l}$ for all k and l must be treated as known. We will consider typical random realizations of $\mathbf{Q}_{k,l}$ for all k and l, which exclude those (such as $\mathbf{Q}_{k,l} = \mathbf{Q}_{k',l'}$ for some $k' \neq k$ or $l' \neq l$) that would occur with extremely small probability.

To understand the correlation between $\mathbf{x} \in \mathcal{S}^{N-1}(1)$ and $\mathbf{u}_{k,x,1} \in \mathcal{S}^{N-1}(1)$ subject to a fixed set of $\mathbf{Q}_{k,l}$, we consider the following

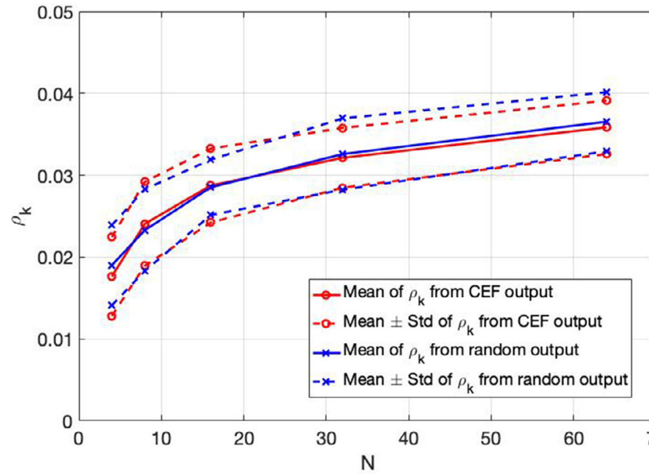

Fig. 3. The means and means±deviations of ρ_k (using SVD-CEF output) and ρ_k^* (using random output) versus N subject to $\eta_{k,x} < 2.5$.

Table 8
Maximums of absolute normalized correlations among the outputs of CEFs.

$\mathbf{x}$	SVD-CEF	IoM-2	IoM-1	DRP	URP
0.0085	0.012	0.21	0.25	0.49	0.81

measure:

$$\rho_k = N \max_{i,j} |[\mathcal{E}_x\{\mathbf{x}\mathbf{u}_{k,x,1}^T\}]_{i,j}| \tag{63}$$

where $\mathcal{E}_x$ denotes the expectation over the distribution of $\mathbf{x}$. If $\mathbf{u}_{k,x,1} = \mathbf{x}$, then $\rho_k = 1$. So, if the correlation between $\mathbf{x}$ and $\mathbf{u}_{k,x,1}$ is small, so should be ρ_k. For comparison, we define ρ_k^* as ρ_k with $\mathbf{u}_{k,x,1}$ replaced by a random unit-norm vector (independent of $\mathbf{x}$).

For a different k, there is a different realization of $\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}$. Hence, ρ_k changes with k. Shown in Fig. 3 are the mean and mean±deviation of ρ_k and ρ_k^* versus N subject to $\eta_{k,x} < 2.5$. We used 10000×100 realizations of $\mathbf{x}$ and $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$. We see that ρ_k and ρ_k^* have virtually the same mean and deviation. (Without the constraint $\eta_{k,x} < 2.5$, ρ_k and ρ_k^* match even better with each other.) In other words, the correlation between the input and output of SVD-CEF is virtually the same as the correlation between the (unit-norm) input of SVD-CEF and an (unit-norm) random vector.

6.2.2. Correlations among the output samples

We now consider the correlation among $y_k = f_k(\mathbf{x})$ for $k = 1, \cdots, K$ of SVD-CEF subject to $\mathbf{x}$ being $\mathcal{N}(0, \mathbf{I}_N)$ and a typical realization of $\mathbf{Q}_{k,l}$ for $k = 1, \cdots, K$ and $l = 1, \cdots, N$. We define the following normalized sample covariance/correlation matrix:

$$\mathbf{C}_{\texttt{SVD-CEF},R} = N\mathcal{E}_{\mathbf{x},R}\{\mathbf{y}_{\texttt{SVD-CEF}}\mathbf{y}_{\texttt{SVD-CEF}}^T\} \tag{64}$$

where $\mathbf{y}_{\texttt{SVD-CEF}} = [y_1, \cdots, y_K]^T$ with its kth entry y_k being the first entry of $\mathbf{u}_{k,x,1}$, and $\mathcal{E}_{\mathbf{x},R}$ denotes the sample average over R realizations of $\mathbf{x}$ (which treats all other quantities such as key-dependent matrices as fixed). We also define $\mathbf{C}_{\texttt{URP},R} = \mathcal{E}_{\mathbf{x},R}\{\mathbf{y}_{\texttt{URP}}\mathbf{y}_{\texttt{URP}}^T\}$ with $\mathbf{y}_{\texttt{URP}}$ being a vertical stack of $\mathbf{y}_k$ in (20) for $k = 1, \cdots, K_0$ with $NK_0 = K$. Similarly, we let $\mathbf{C}_{\texttt{DRP},R} = c_{\texttt{DRP}}\mathcal{E}_{\mathbf{x},R}\{\mathbf{y}_{\texttt{DRP}}\mathbf{y}_{\texttt{DRP}}^T\}$ and $\mathbf{C}_{\texttt{IoM},R} = c_{\texttt{IoM}}\mathcal{E}_{\mathbf{x},R}\{\mathbf{y}_{\texttt{IoM}}\mathbf{y}_{\texttt{IoM}}^T\}$ where $c_{\texttt{DRP}}$ and $c_{\texttt{IoM}}$ are such that the diagonal elements of each of $\mathbf{C}_{\texttt{DRP},R}$ and $\mathbf{C}_{\texttt{IoM}}$ have their averaged value equal to one. For IoM, each entry of $\mathbf{y}_{\texttt{IoM}}$ is an integer "index-of-max" (ranging from 0 to $N-1$) minus $\frac{N-1}{2}$, which ensures that each entry of $\mathbf{y}_{\texttt{IoM}}$ has the zero mean.

Shown in Table 8 are the maximum value of the absolute off-diagonal elements of each of the above defined sample covariance matrices with $N = 16$, $K = 128$ and $R = 10^5$. The first column in

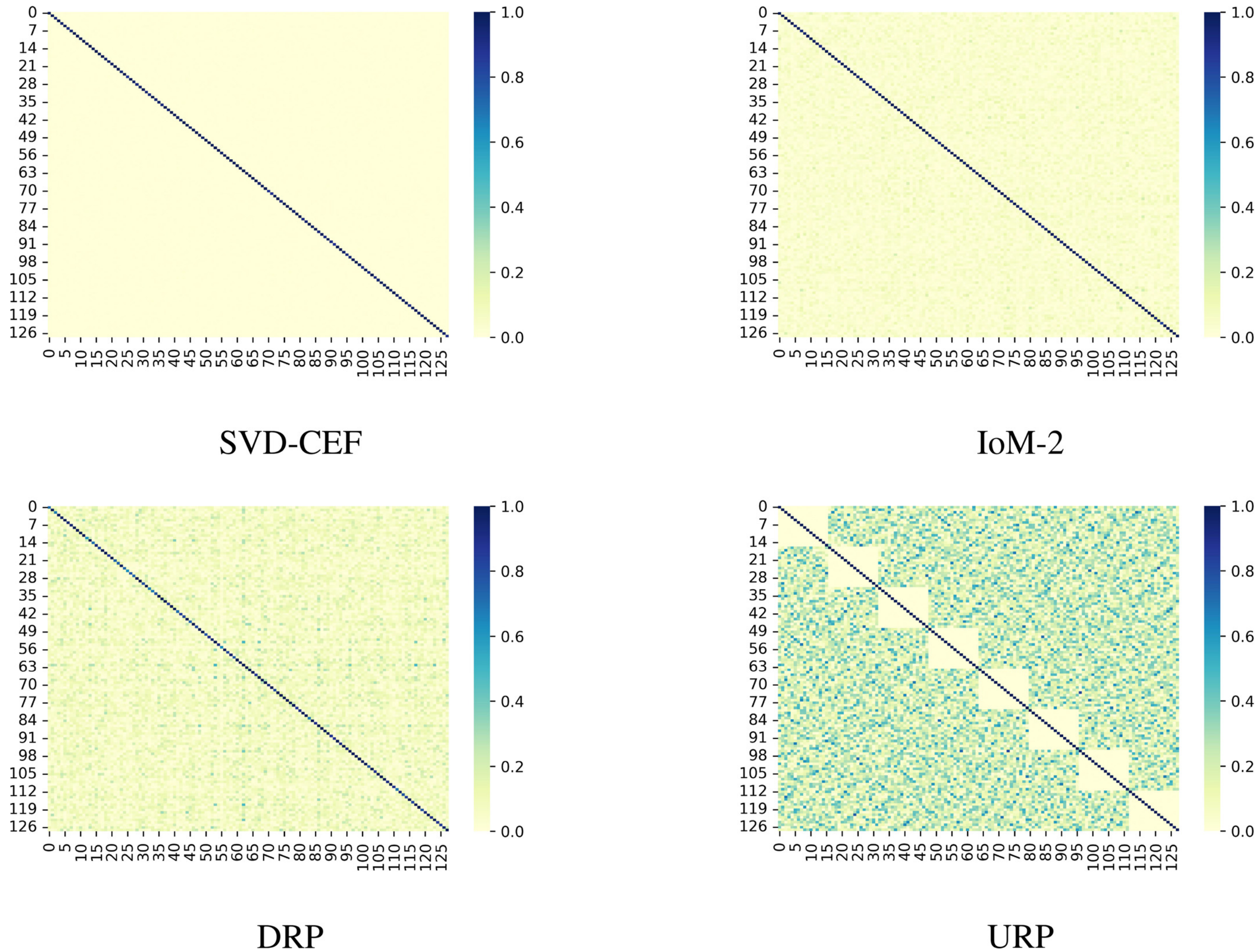

Fig. 4. Correlation “heatmaps” of the output samples of SVD-CEF, IoM-2, DRP and URP (when there is no secret key used in any of these CEFs).

Table 8 is for $\mathbf{C}_{\mathbf{x},R} = \mathcal{E}_{\mathbf{x},R}\{\mathbf{x}\mathbf{x}^T\}$ of $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_N)$, which serves as a reference. We know that as $R \to \infty$, the peak sample correlation of the elements in $\mathbf{x}$ goes to zero. (The mean and deviation of each off-diagonal element of $\mathbf{C}_{\mathbf{x},R}$ are zero and $\frac{1}{\sqrt{R}}$, respectively. At $R = 10^5$, $\frac{1}{\sqrt{R}} = 0.0032$.) We see that the peak sample correlation of SVD-CEF is very small and comparable to (about 1.4 times) that of $\mathbf{x}$. On the other hand, the peak sample correlations of IoM, DRP and URP are about 17 to 67 times larger than that of SVD-CEF. We should stress that the values in this table will change, but only slightly with high probability, if different realizations of the random matrices and/or operations in the CEFs are used.

Illustrated in Fig. 4 are the “heatmaps” of the absolute values of the entries of the sample covariance matrices of SVD-CEF, IoM-2, DRP and URP, where all parameters are the same as those for Table 8. Each of these heatmaps is based on a random realization of their embedded pseudorandom transformations. However, the overall patterns of the heatmaps in general do not change much as these pseudorandom transformations are chosen differently. We see that the output samples of SVD-CEF have virtually zero correlations, which in fact do not differ much from the sample correlations of the entries in $\mathbf{x}$. This is because of the unique relationship between the principal eigenvector $\mathbf{u}_{k,x,1}$ of $\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T$ and the input vector $\mathbf{x}$. We also see that most of the correlations of IoM-2 are also small but not as small as those of SVD-CEF. And there are still a lot of scattered “peaks” in the heatmap of IoM-2, which are quite significant. The heatmaps of DRP and URP show overwhelmingly large correlation values. For URP, the sample correlations among samples within each subvector $\mathbf{y}_k$ are small in the order of $\frac{1}{\sqrt{R}}$, which is due to unitary transformation. But the correlation between $\mathbf{y}_k$ and $\mathbf{y}_l$ for $k \neq l$ is rather large as shown in this figure, which is because of the linear nature of URP and the non-orthogonality among any set of L N-dimensional vectors with $L > N$. For the same reason, RP proposed in [8] also has a very poor property in correlation.

6.3. Invariance

We show next via simulation that $\mathbf{u}_{k,x,1}$ for each k is nearly uniformly distributed on $\mathcal{S}^{N-1}(1)$ when $\mathbf{x}$ is uniformly distributed on $\mathcal{S}^{N-1}(1)$, which implies that y_k of SVD-CEF for each k has the same distribution (i.e., invariant to k).

To show that the distribution of $\mathbf{u}_{k,x,1}$ for each k is nearly uniform on $\mathcal{S}^{N-1}(1)$, we show that for any k and any unit-norm vector $\mathbf{v}$, the probability density function (PDF) $p_{k,v}(x)$ of $\mathbf{v}^T\mathbf{u}_{k,x,1}$ subject to a fixed set of $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$ and a uniform random $\mathbf{x}$ on $\mathcal{S}^{N-1}(1)$ is nearly the same as the PDF $p(x)$ of an element in $\mathbf{x}$. The expression of $p(x)$ is derived in (83) in Appendix B. The distance

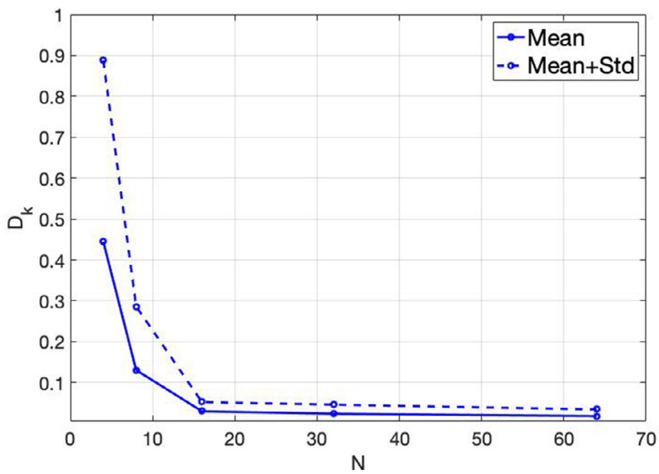

Fig. 5. The mean and mean±deviation of $D_{k,v}$ versus N subject to $\eta_{k,x} < 2.5$.

between $p(x)$ and $p_{k,v}(x)$ can be measured by

$$D_{k,v} = \int p(x) \ln \frac{p(x)}{p_{k,v}(x)} dx \geq 0. \tag{65}$$

Clearly, $D_{k,v}$ changes as k and $\mathbf{v}$ change. Shown in Fig. 5 are the mean and mean ± deviation of $D_{k,v}$ versus N subject to $\eta_{k,x} < 2.5$. We used $50 \times 1000 \times 500$ realizations of $\mathbf{v}$, $\mathbf{x}$ and $\{\mathbf{Q}_{k,1}, \cdots, \mathbf{Q}_{k,N}\}$. We see that $D_{k,v}$ becomes very small as N increases beyond 15. This means that for a moderate or large N, $\mathbf{u}_{k,x,1}$ is (at least approximately) uniformly distributed on $\mathcal{S}^{N-1}(1)$ when $\mathbf{x}$ is uniformly distributed on $\mathcal{S}^{N-1}(1)$. (Without the constraint $\eta_{k,x} < 2.5$, $D_{k,v}$ versus N has a similar pattern and is even slightly smaller.) In other words, for a moderate or large N, the output sample y_k of SVD-CEF for each k has a PDF approximately given by (83) in Appendix B, which is invariant to k.

7. Further comparison between SVD-CEF and IoM-2

As discussed earlier, the per (integer) sample complexity of forward computation of IoM-2 is $\mathcal{O}(N^2)$ while the per (real) sample complexity of forward computation of SVD-CEF is $\mathcal{O}(N^3)$. And the best known method to attack IoM-2 has the complexity $L_{N,M}2^N$ with $L_{N,M}$ being a linear function of M and N respectively while the best known method to attack SVD-CEF has the complexity $P_{N,M}2^{\zeta N}$ with $\zeta > 1$ increasing with N and $P_{N,M}$ being a polynomial function of M and N. Furthermore, SVD-CEF has much smaller output correlations than IoM-2.

Note that while SVD-CEF is much harder to attack than IoM-2, none of the two could be shown yet to be easy to attack (assuming that all elements in $\mathbf{x}$ have independently random signs from the perspective of the attacker). In this regard, both SVD-CEF and IoM-2 somewhat stand out among all the CEFs considered in this paper.

We will next compare the noise sensitivities of SVD-CEF and IoM-2. To do so, we need to quantize the output of SVD-CEF as shown below since the output of IoM-2 is always discrete.

7.1. Quantization of SVD-CEF

Let the kth (real-valued) sample of the output of SVD-CEF at Alice due to the input vector $\mathbf{x}$ be y_k, and the kth sample of the output of SVD-CEF at Bob due to the input vector $\mathbf{x}' = \mathbf{x} + \mathbf{w}$ be y'_k. In the simulation, we will assume that the perturbation vector $\mathbf{w}$ is white Gaussian, i.e., $\mathcal{N}(0, \sigma_w^2 \mathbf{I})$.

As shown before, the PDF of y_k can be approximated by (83) in Appendix B, i.e., $f_{y_k}(y) = C_N(1-y^2)^{\frac{N-3}{2}}$ with $C_N = \frac{\Gamma(\frac{N}{2})}{\sqrt{\pi}\Gamma(\frac{N-1}{2})}$ and $-1 < y < 1$. To quantize y_k into $b_y = \log_2 B_y$ bits, Alice first over quantizes y_k into $m_y = \log_2 M_y$ bits with $M_y = B_y L_y$. Each of the M_y quantization intervals within $(-1, 1)$ is chosen to have the same probability $\frac{1}{M_y}$. For example, the left-side boundary value t_i of the ith interval can be computed (offline) from $\int_{-1}^{t_i} f_{y_k}(y)dy = \frac{i}{M_y}$ with $i = 0, 1, \cdots, M_y - 1$. A closed form of $\int (1-y^2)^{\frac{N-3}{2}} dy = \int \cos^{N-2}\theta d\theta$ with $y = \sin\theta$ is available for efficient bisection search of t_i. Specifically, $\int \cos^n\theta d\theta = \frac{\cos^{n-1}\theta \sin\theta}{n} + \frac{n-1}{n}\int \cos^{n-2}\theta d\theta$.

The additional $l_y = \log_2 L_y$ bits are used to assist the quantization of y'_k at Bob. Specifically, if y_k is quantized by Alice into an integer $0 \leq i_k \leq M_y - 1$, which has the standard binary form $d_1 \cdots d_{b_y} d_{b_y+1} \cdots d_{m_y}$, then Alice keeps the first b_y bits $d_1 \cdots d_{b_y}$, corresponding to an integer $0 \leq m_k \leq B_y - 1$, and informs Bob of the last l_y bits $d_{b_y+1} \cdots d_{m_y}$, corresponding to an integer $0 \leq j_k \leq L_y - 1$. Then the quantization of y'_k by Bob is $m'_k = arg\min_{m=0,\cdots,B_y-1} |y'_k - j_k - mL_y|$.

If m_k differs from m'_k, it is very likely that $m'_k = m_k \pm 1$. So, Gray binary code should be used to represent the integers m_k and m'_k at Alice and Bob respectively. If $m'_k = m_k \pm 1$, Gray codes of m_k and m'_k only differ from each other by one bit.

The above quantization scheme is related to those for secret key generation in [15] and [16]. Here, we have a virtually unlimited amount of y_k and y'_k for $k \geq 1$. A limited bit error rate after quantization is not a problem in such applications as biometrics based authentication (where “Alice” corresponds to “registration phase” and “Bob” “validation phase”).

7.2. Comparison of Bit Error Rates

We next compare the bit error rates (BERs) between the quantized SVD-CEF and IoM-2. For each pair of $\mathbf{x}$ and $\mathbf{x}' = \mathbf{x} + \mathbf{w}$, we will assume that SVD-CEF and IoM-2 each produces a pair of sequences each of at least L_{key} bits.

Furthermore, we assume that for each of $k = 1, \cdots, K_2$, IoM-2 applies N random permutations to the $N \times 1$ feature vector $\mathbf{x}$ at Alice to produce $\mathbf{v}_{k,1}, \cdots, \mathbf{v}_{k,N}$ respectively, and then computes the element-wise products of these vectors to produce $\mathbf{h}_k$. The index of the largest entry in $\mathbf{h}_k$ is now denoted by $0 \leq m_k \leq N-1$, which corresponds to a string of $\log_2 N$ binary bits for Alice. Bob conducts the same operations on $\mathbf{x}' = \mathbf{x} + \mathbf{w}$ to produce $0 \leq m'_k \leq N-1$, which corresponds to a string of $\log_2 N$ binary bits for Bob. We also apply Gray binary code here for IoM-2, which however has little effect on the performance. For each fixed pair of $\mathbf{x}$ and $\mathbf{x}'$, the above process is repeated (with independent sets of permutations) for all $k = 1, \cdots, K_2$, which yields a pair of binary sequences each of $K_2 \log_2 N \geq L_{key}$ bits. With R random realizations of $\mathbf{x}$ and $\mathbf{x}'$ (and the corresponding set of random permutations), the above process yields a pair of sequences each of $RK_2 \log_2 N$ bits, from which the BER of IoM-2 is computed. Namely, the BER is estimated by $\frac{1}{RK_2 \log_2 N}$ times the number of mismatched bits in the two sequences.

For each pair of $\mathbf{x}$ and $\mathbf{x}'$, the quantized SVD-CEF first generates $y_1, \cdots, y_K$ (based on $\mathbf{x}$) for Alice and $y'_1, \cdots, y'_K$ (based on $\mathbf{x}' = \mathbf{x} + \mathbf{w}$) for Bob, which are then quantized into a pair of sequences each of $Kb_y \geq L_{key}$ bits where b_y is the number of bits per output sample of SVD-CEF. With R realizations of $\mathbf{x}$ and $\mathbf{x}'$ (and the corresponding realizations of $\mathbf{Q}_{k,l}$ for $1 \leq k \leq K$ and $1 \leq l \leq N$), the quantized SVD-CEF also yields a pair of sequences each of RKb_y bits, from which BER is computed.

We consider two choices of b_y, i.e., $b_y = \log_2 N$ and $b_y = 1$. The first choice means that each output sample of SVD-CEF yields the same number of bits as that of IoM-2. But the second choice yields one bit per output sample of SVD-CEF. By reducing the number of bits per sample, we can reduce the BER significantly for SVD-CEF.

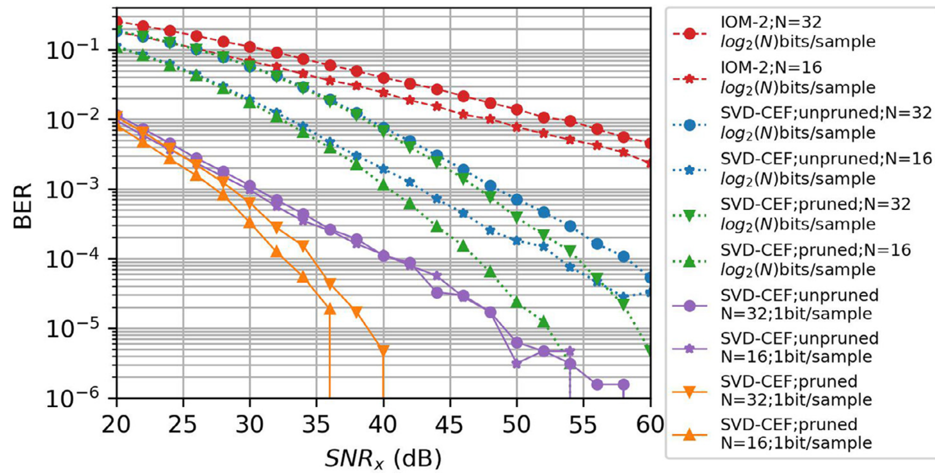

Fig. 6. A BER comparison of quantized SVD-CEF and IoM-2. The vertical line at the end of a curve indicates that the next value is below 10^{-6}.

The computation cost of the second choice is only increased by a factor no more than $\log_2 N$, which is not very significant. With a reduced b_y, the quantization complexity is also reduced. This option is not available for IoM-2. If we constrain the search of the index-of-max among the first $L < N$ elements in $\mathbf{h}_k$, it only reduces the number of bits per output sample but does not improve the BER of IoM-2. This is because all entries in each $\mathbf{h}_k$ are statistically the same.

In Fig 6, we compare the BER performances of the quantized SVD-CEF and IoM-2, where $L_{key} = 128$, $K_2 = \left\lceil \frac{L_{key}}{\log_2 N} \right\rceil$, $K = \left\lceil \frac{L_{key}}{b_y} \right\rceil$, $R = 5000$, $\mathbf{x} \sim \mathcal{N}(0, \mathbf{I}_N)$, $\mathbf{w} \sim \mathcal{N}(0, \sigma_w^2 \mathbf{I}_N)$ and $\mathtt{SNR}_x = \frac{1}{\sigma_w^2}$. In the figure, we considered all combinations of $N = 16$ vs $N = 32$, $b_y = \log_2 N$ vs $b_y = 1$ for SVD-CEF, and pruned vs unpruned SVD-CEF. In the case of pruning, we used $\eta_{k,x} < 2.5$. We see that IoM-2 is outperformed significantly by SVD-CEF with or without pruning for both cases of b_y. As expected, using $b_y = 1$ (instead of $b_y = \log_2 N$), SVD-CEF has a dramatic (several orders of magnitude) reduction of BER. The somewhat irregular pattern of BER vs SNR, when BER is very small, is due to the limited number R of runs used in the simulation.

8. Conclusion

In this paper, we have presented a systematic development of continuous encryption functions (CEFs) that transcend the boundaries of wireless network science and biometric data science. The development of CEFs is critically important for physical layer encryption of wireless communications and biometric template security for online Internet applications among others. While the family of CEFs defined in this paper include all prior continuous one-way functions, we proposed a list of criteria for a good CEF desirable in applications, which are the hardness to invert, the hardness to substitute, the sensitivity to noise, the correlation among the output samples and the invariance of the output distributions. We showed that the dynamic random projection (DRP) method and the index-of-max hashing algorithm 1 (IoM-1) are not hard to invert, the index-of-max hashing algorithm 2 (IoM-2) is not as hard to invert as it was thought to be, and the higher-order polynomials (HOP) method is easy to attack via substitution. We also showed that DRP and IoM have relatively poor properties in terms of their output correlations, and HOP is highly sensitive to noise. We have introduced a singular value decomposition (SVD) based CEF, which is shown empirically to be hard to attack. Our statistical analyses and simulation results also verified that SVD-CEF has relatively good properties in its noise sensitivity, its output correlation and the invariance of its output distribution. Despite their lower complexity in forward computation, none of the prior continuous one-way functions reviewed in this paper is able to compete against SVD-CEF favorably under the five criteria proposed in this paper. However, if there is already a strong secret key, the unitary random projection (URP) discussed in this paper should be the first to consider in applications.

During the review of this paper, speculation of alternative approaches such as (higher-order) tensors and chaos systems was raised. It is unknown right now whether these or other approaches could lead to a better CEF than SVD-CEF. The hardness to invert or substitute is only part of the requirements for a good CEF.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Yingbo Hua reports a relationship with University of California Riverside that includes: employment. Yingbo Hua has patent pending to University of California.

Appendix A. Attack of SVD-CEF via EVD Equilibrium in X

We show next the details of an attack algorithm based on (51). Similar attack algorithms developed from (50) and (52) are omitted. An earlier result was also reported in [2].

It is easy to verify that $\mathbf{X} = \alpha \mathbf{I}_N + (1-\alpha)\mathbf{x}\mathbf{x}^T$ with any $-\infty < \alpha < \infty$ is a solution to the following

$$\left(\sum_{l=1}^{N} \mathbf{Q}_{k,l} \mathbf{X} \mathbf{Q}_{k,l}^T\right) \mathbf{u}_{k,x,1} = c_{k,x,1} \mathbf{u}_{k,x,1} \tag{66}$$

where $c_{k,x,1} = \alpha + (1-\alpha)\sigma_{k,x,1}^2$. The expression (66) is more precise and more revealing than (51) for the desired unknown matrix $\mathbf{X}$.

To ensure that $\mathbf{u}_{k,x,1}$ from (66) is unique, it is sufficient and necessary to find a $\mathbf{X}$ with the above structure and $1-\alpha \neq 0$. To ensure $1-\alpha \neq 0$, we assume that $x_1 x_2 \neq 0$ where x_1 and x_2 are the first two elements of $\mathbf{x}$. Then we add the following constraint:

$$(\mathbf{X})_{1,2} = (\mathbf{X})_{2,1} = 1. \tag{67}$$

which is in addition to the previous condition $Tr(\mathbf{X}) = 1$. Now for the expected solution structure $\mathbf{X} = \alpha \mathbf{I}_N + (1-\alpha)\mathbf{x}\mathbf{x}^T$, we have $1 - \alpha = \frac{1}{x_1 x_2} \neq 0$.

Note that $c_{k,x,1}$ in (66) is either the largest or the smallest eigenvalue of $\sum_{l=1}^{N}\mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T$ corresponding to whether $1-\alpha$ is positive or negative.

To develop the Newton's algorithm, we now take the differentiation of (66) to yield

$$\left(\sum_{l=1}^{N}\mathbf{Q}_{k,l}\partial\mathbf{X}\mathbf{Q}_{k,l}^T\right)\mathbf{u}_{k,x,1}+\left(\sum_{l=1}^{N}\mathbf{Q}_{k,l}\mathbf{X}\mathbf{Q}_{k,l}^T\right)\partial\mathbf{u}_{k,x,1}$$
$$=\partial c_k\mathbf{u}_{k,x,1}+c_k\partial\mathbf{u}_{k,x,1} \tag{68}$$

where we have used $\mathbf{u}_{k,x,1}=\mathbf{u}_{k,x,1}$ and $c_k=c_{k,x,1}$ for convenience. The first term is equivalent to $\tilde{\mathbf{Q}}_k\partial\tilde{\mathbf{x}}$ with $\tilde{\mathbf{Q}}_k=(\sum_{l=1}^{N}\mathbf{u}_{k,x,1}^T\mathbf{Q}_{k,l}\otimes\mathbf{Q}_{k,l})$ and $\tilde{\mathbf{x}}=vec(\mathbf{X})$. (For basics of matrix differentiation, see [20].)

Since $\mathbf{X}=\mathbf{X}^T$, there are repeated entries in $\tilde{\mathbf{x}}$. We can write $\tilde{\mathbf{x}}=[\tilde{\mathbf{x}}_1^T,\cdots,\tilde{\mathbf{x}}_N^T]^T$ with $\tilde{\mathbf{x}}_n=[\tilde{x}_{n,1},\cdots,\tilde{x}_{n,N}]^T$ and $\tilde{x}_{i,j}=\tilde{x}_{j,i}$ for all $i\neq j$. Let $\hat{\mathbf{x}}$ be the vectorized form of the lower triangular part of $\mathbf{X}$. Then it follows that

$$\tilde{\mathbf{Q}}_k\partial\tilde{\mathbf{x}}=\hat{\mathbf{Q}}_k\partial\hat{\mathbf{x}} \tag{69}$$

where $\hat{\mathbf{Q}}_k$ is a compressed form of $\tilde{\mathbf{Q}}_k$ as follows. Let $\tilde{\mathbf{Q}}_k=[\tilde{\mathbf{Q}}_{k,1},\cdots,\tilde{\mathbf{Q}}_{k,N}]$ with $\tilde{\mathbf{Q}}_{k,n}=[\tilde{\mathbf{q}}_{k,n,1},\cdots,\tilde{\mathbf{q}}_{k,n,N}]$. For all $1\leq i<j\leq N$, replace $\tilde{\mathbf{q}}_{k,i,j}$ by $\tilde{\mathbf{q}}_{k,i,j}+\tilde{\mathbf{q}}_{k,j,i}$, and then drop $\tilde{\mathbf{q}}_{k,j,i}$. The resulting matrix is $\hat{\mathbf{Q}}_k$.

The differential of $Tr(\mathbf{X})=1$ is $Tr(\partial\mathbf{X})=0$ or equivalently $\mathbf{t}^T\partial\hat{\mathbf{x}}=0$ where $\mathbf{t}^T=[\mathbf{t}_1^T,\cdots,\mathbf{t}_N^T]$ and $\mathbf{t}_n^T=[1,\mathbf{0}_{1\times(N-n)}]^T$.

Combining the above for all k along with $\mathbf{u}_{k,x,1}^T\partial\mathbf{u}_{k,x,1}=0$ (due to the norm constraint $\|\mathbf{u}_{k,x,1}\|^2=1$) for all k, we have

$$\mathbf{A}_x\partial\hat{\mathbf{x}}+\mathbf{A}_u\partial\mathbf{u}+\mathbf{A}_z\partial\mathbf{z}=0 \tag{70}$$

where

$$\mathbf{A}_x=\begin{bmatrix}\mathbf{t}^T\\ \hat{\mathbf{Q}}_1\\ \cdots\\ \hat{\mathbf{Q}}_K\\ \mathbf{0}_{K\times\frac{1}{2}N(N+1)}\end{bmatrix} \tag{71}$$

$$\mathbf{A}_u=\begin{bmatrix}\mathbf{0}_{1\times NK}\\ diag(\mathbf{G}_{1,x},\cdots,\mathbf{G}_{K,x})\\ diag(\mathbf{u}_1^T,\cdots,\mathbf{u}_K^T)\end{bmatrix}, \tag{72}$$

$$\mathbf{A}_z=\begin{bmatrix}\mathbf{0}_{1\times K}\\ -diag(\mathbf{u}_1,\cdots,\mathbf{u}_K)\\ \mathbf{0}_{K\times K}\end{bmatrix} \tag{73}$$

with $\mathbf{G}_{k,x}=\mathbf{M}_{k,x}\mathbf{M}_{k,x}^T-c_k\mathbf{I}_M$.

Now we partition $\mathbf{u}$ into two parts: $\mathbf{u}_a$ (known) and $\mathbf{u}_b$ (unknown). Also partition $\mathbf{A}_u$ into $\mathbf{A}_{u,a}$ and $\mathbf{A}_{u,b}$ such that $\mathbf{A}_u\partial\mathbf{u}=\mathbf{A}_{u,a}\partial\mathbf{u}_a+\mathbf{A}_{u,b}\partial\mathbf{u}_b$. Since $(\mathbf{X})_{1,2}=(\mathbf{X})_{2,1}=1$, we also let $\hat{\mathbf{x}}_0$ be $\hat{\mathbf{x}}$ with its second element removed, and $\mathbf{A}_{x,0}$ be $\mathbf{A}_x$ with its second column removed. It follows from (70) that

$$\mathbf{A}\partial\mathbf{a}+\mathbf{B}\partial\mathbf{b}=0 \tag{74}$$

where $\mathbf{a}=\mathbf{u}_a$, $\mathbf{b}=[\hat{\mathbf{x}}_0^T,\mathbf{u}_b^T,\mathbf{z}^T]^T$, $\mathbf{A}=\mathbf{A}_{u,a}$, $\mathbf{B}=[\mathbf{A}_{x,0},\mathbf{A}_{u,b},\mathbf{A}_z]$.

Based on (74), the Newton's algorithm is

$$\left[\hat{\mathbf{x}}_0^{(i+1)}\right]=\left[\hat{\mathbf{x}}_0^{(i)}\right]-\eta(\mathbf{B}^T\mathbf{B})^{-1}\mathbf{B}^T\mathbf{A}(\mathbf{u}_a-\mathbf{u}_a^{(i)}) \tag{75}$$

where the terms associated with $*$ are not needed, $\mathbf{u}_a^{(i)}$ is the ith-step "estimate" of the known vector $\mathbf{u}_a$ (through forward computation) based on the i-step estimate $\hat{\mathbf{x}}_0^{(i)}$ of the unknown vector $\hat{\mathbf{x}}_0$. This algorithm requires $NyK\geq\frac{1}{2}N(N+1)-1$ in order for $\mathbf{B}$ to have full column rank.

For a random initialization around $\mathbf{X}$, we can let $\mathbf{X}'=(1-\beta)\mathbf{X}+\beta\mathbf{W}$ where $\mathbf{W}$ is a symmetric random matrix with $Tr(\mathbf{W})=1$. Furthermore, $(\mathbf{W})_{1,2}=(\mathbf{W})_{2,1}$ is such that $(\mathbf{X}')_{1,2}=(\mathbf{X}')_{2,1}=1$. Keep in mind that at every step of iteration, we keep $(\mathbf{X}^{(i)})_{1,2}=(\mathbf{X}^{(i)})_{2,1}=1$.

Upon convergence of $\mathbf{X}$, we can also update $\mathbf{x}$ as follows. Let the eigenvalue decomposition of $\mathbf{X}$ be $\mathbf{X}=\sum_{i=1}^{N}\lambda_i\mathbf{e}_i\mathbf{e}_i^T$ where $\lambda_1>\lambda_2>\cdots>\lambda_N$. Then the update of $\mathbf{x}$ is given by $\mathbf{e}_1$ if $1-\alpha>0$ or by $\mathbf{e}_N$ if $1-\alpha<0$. With each renewed $\mathbf{x}$, there are a renewed α and hence a renewed $\mathbf{X}$ (i.e., by setting $\mathbf{X}=\alpha\mathbf{I}+(1-\alpha)\mathbf{x}\mathbf{x}^T$ with $1-\alpha=\frac{1}{x_1x_2}$). Using the new $\mathbf{X}$ as the initialization, we can continue the search using (75).

The performance of the algorithm (75) is discussed in Section 5.2.

Appendix B. Distributions of Elements of a Uniformly Random Vector on Sphere

Let $\mathbf{x}$ be uniformly random on $\mathcal{S}^{n-1}(r)$. This vector can be parameterized as follows:

$$\begin{aligned}x_1&=r\cos\theta_1\\ x_2&=r\sin\theta_1\cos\theta_2\\ &\cdots\\ x_{n-1}&=r\sin\theta_1\cdots\sin\theta_{n-2}\cos\theta_{n-1}\\ x_n&=r\sin\theta_1\cdots\sin\theta_{n-2}\sin\theta_{n-1}\end{aligned}$$

where $0<\theta_i\leq\pi$ for $i=1,\cdots,n-2$, and $0<\theta_{n-1}\leq 2\pi$. According to Theorem 2.1.3 in [22], the differential of the surface area on $\mathcal{S}^{n-1}(r)$ is

$$dS^{n-1}(r)=r^{n-1}\sin^{n-2}\theta_1\sin^{n-3}\theta_2\cdots\sin\theta_{n-2}d\theta_1\cdots d\theta_{n-1} \tag{76}$$

We know that $\int_{\mathcal{S}^{n-1}(r)}dS^{n-1}(r)=|\mathcal{S}^{n-1}(r)|=\frac{2\pi^{n/2}}{\Gamma(\frac{n}{2})}r^{n-1}$. Hence, the PDF of $\mathbf{x}$ is

$$f_x(\mathbf{x})=\frac{1}{|\mathcal{S}^{n-1}(r)|}. \tag{77}$$

B0.1. Distribution of one element in $\mathbf{x}$

We can rewrite $\int_{\mathcal{S}^{n-1}(r)}f_x(\mathbf{x})dS^{n-1}(r)=1$ as

$$\int_{\theta_1}\left[\int_{\mathcal{S}^{n-2}(r\sin\theta_1)}f_x(\mathbf{x})rdS^{n-2}(r\sin\theta_1)\right]d\theta_1=1 \tag{78}$$

or equivalently

$$\int_{\theta_1}\left[\frac{|\mathcal{S}^{n-2}(r\sin\theta_1)|}{|\mathcal{S}^{n-1}(r)|}r\right]d\theta_1=1. \tag{79}$$

Hence the PDF of θ_1 is

$$f_{\theta_1}(\theta_1)=\frac{|\mathcal{S}^{n-2}(r\sin\theta_1)|}{|\mathcal{S}^{n-1}(r)|}r. \tag{80}$$

To find the PDF of $x_1=r\cos\theta_1$, we have

$$f_{x_1}(x_1)=f_{\theta,1}(\theta_1)\frac{1}{\left|\frac{dx_1}{d\theta_1}\right|}=\frac{f_{\theta,1}(\theta_1)}{|r\sin\theta_1|} \tag{81}$$

where $r\sin\theta_1=\sqrt{r^2-x_1^2}$. Therefore, combining all the previous results yields

$$f_{x_1}(x_1)=\frac{\Gamma(\frac{n}{2})}{\sqrt{\pi}\Gamma(\frac{n-1}{2})}\frac{(r^2-x_1^2)^{\frac{n-3}{2}}}{r^{n-2}} \tag{82}$$

where $-r\leq x_1\leq r$.

If $r=1$, we have

$$f_{x_1}(x_1)=\frac{\Gamma(\frac{n}{2})}{\sqrt{\pi}\Gamma(\frac{n-1}{2})}(1-x_1^2)^{\frac{n-3}{2}} \tag{83}$$

where $-1\leq x_1\leq 1$. This is the PDF $p(x)$ in Section 6.3.

Due to symmetry, we know that x_i for any i has the same PDF as x_1. Also note that if $n = 3$, $f_{x_1}(x)$ is a uniform distribution.

References

[1] Y. Hua, Reliable and secure transmissions for future networks, IEEE ICASSP'2020 (2020) 2560–2564.
[2] Y. Hua, A. Maksud, Unconditional secrecy and computational complexity against wireless eavesdropping, IEEE SPAWC'2020 (2020) 5.
[3] A. Maksud, Y. Hua, Physical layer encryption for UAV-to-ground communications, IEEE ICC'22 Workshop - UAV5G (2022).
[4] A. Maksud, Y. Hua, Secret key generation by continuous encryption before quantization, IEEE Signal Process Lett 29 (2022) 1497–1501.
[5] S. Wu, Y. Hua, Total secrecy from anti-eavesdropping channel estimation, IEEE Trans. Signal Process. 70 (2022) 1088–1103.
[6] A.K. Jain, K. Nandakumar, A. Nagar, Biometric template security, EURASIP J Adv Signal Process (2008).
[7] D.V.M. Patel, N.K. Ratha, R. Chellappa, Cancelable biometrics, IEEE Signal Process Mag (2015).
[8] A.B.J. Teoh, C.T. Young, Cancelable biometrics realization with multispace random projections, IEEE Transactions on Systems, Man and Cybernetics 37 (5) (2007) 1096–1106.
[9] E.B. Yang, D. Hartung, K. Simoens, C. Busch, Dynamic random projection for biometric template protection, Proc. IEEE Int. Conf. Biometrics: Theory Applications and Systems (2010) 1–7.
[10] D. Grigoriev, S. Nikolenko, Continuous hard-to-invert functions and biometric authentication, Groups 44 (1) (2012) 19–32.
[11] Z. Jin, Y.-L. Lai, J.Y. Hwang, S. Kim, A.B.J. Teoh, Ranking based locality sensitive hashing enabled cancelable biometrics: index-of-max hashing, IEEE Transactions on Information Forensic and Security 13 (2) (2018).
[12] J.K. Pillai, V.M. Patel, R. Chellappa, N.K. Ratha, Secure and robust iris recognition using random projections and sparse representation, IEEE Trans Pattern Anal Mach Intell 33 (9) (2011).
[13] S. Kirchgasser, C. Kauba, Y.-L. Lai, J. Zhe, A. Uhl, Finger vein template protection based on alignment-robust feature description and index-of-maximum hashing, IEEE Transactions on Biometrics, Behavior, and Identity Science 2 (4) (2020) 337–349.
[14] L. Lai, S.-W. Ho, H.V. Poor, Privacy-security trade-offs in biometric security systems - Part I: single use case, IEEE Trans. Inf. Forensics Secur. 6 (1) (2011) 122–139.
[15] J.W. Wallace, R.K. Sharma, Automatic secret keys from reciprocal MIMO wireless channels: measurement and analysis, IEEE Trans. Inf. Forensics Secur. 5 (3) (2010) 381–392.
[16] C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, N.B. Mandayam, Information-theoretically secret key generation for fading wireless channels, IEEE Trans. Inf. Forensics Secur. 5 (2) (2010) 240–254.
[17] L.A. Levin, The tale of one-way functions, arXiv:cs/0012023v5 (2003).
[18] J. Katz, Y. Lindell, Introduction to Modern Cryptography, 2nd ed, CRC, 2015.
[19] G.H. Golub, C.F. Van, Loan, Matrix Computations, John Hopkins University Press, 1983.
[20] J.R. Magnus, H. Neudecker, Matrix Differential Calculus with Applications in Statistics and Econometrics, Wiley, 2002.
[21] A. Greenbaum, R.-C. Li, M.L. Overton, First-order perturbation theory for eigenvalues and eigenvectors, arXiv:1903.00785v2 (2019).
[22] R.J. Muirhead, Aspects of Multivariate Statistical Theory, Wiley, 1982.
[23] T. Gowers, The Princeton Companion to Mathematics, Princeton University Press, 2008.
[24] Y. Hua, A. Maksud, Continuous encryption functions for security over networks, 2021, https://arxiv.org/abs/2111.03163.